

UNIVERSIDAD CARLOS III DE MADRID



Proyecto Fin de Grado

Answer2Pass – Juego de evaluación interactiva de alumnos

Autor: Vicente Domínguez Martín

Tutores: José María de Fuentes García-Romero de Tejada, Jorge Blasco Alís

Titulación: Grado en Ingeniería Informática

15/06/2012

Agradecimientos

Hace cuatro años dio comienzo una nueva etapa en mi vida, la superación de los exámenes de selectividad marcaron el fin del instituto y el comienzo de mis estudios universitarios. Auguré una época dura, donde tendría que emplearme a fondo y trabajar al máximo para superar de forma satisfactoria las dificultades que se estaban por plantear.

Los cuatro años de carrera se sucedieron según lo previsto, un camino lleno de altibajos, interminable en algunas ocasiones, donde el esfuerzo y la constancia son las principales herramientas para conseguir llegar al final. Afortunadamente, personas nuevas en mi vida y en especial otras ya conocidas, se han convertido en un elemento esencial en el recorrido. Por ello quiero agradecerles la labor de haber hecho estos años más fáciles, la ayuda y el apoyo que me han brindado en todo momento, la unión que hemos disfrutado y los momentos y risas memorables que hemos vivido tanto dentro como fuera de la universidad. Dichas nuevas amistades, y las antiguas por la cantidad y buenos momentos vividos, van a formar parte de las mejores cosas que me llevo de la etapa universitaria, consiguiendo verdaderos amigos con los que seguro podré contar en el futuro. También destacar el agradecimiento a otras personas ajenas a la universidad, que me han hecho ver día a día que el final estaba cada vez más cerca.

Por otro lado, quiero agradecer al equipo de programadores de Kantar Health, empresa donde realicé las prácticas, los conocimientos y experiencia que me han proporcionado. En especial a Rubén, que se ha ocupado de aportarme una gran cantidad de conocimientos que me han permitido desarrollar este proyecto y otras prácticas de la carrera de forma muy satisfactoria.

También me gustaría agradecer a Jorge y Chema, tutores del proyecto que he conocido en esta última etapa universitaria, el papel que han desempeñado a lo largo de todo el desarrollo del mismo. Han sido en gran medida los responsables de la finalización satisfactoria del proyecto, ya que me han guiado en todo momento y me han dado ánimos cuando ha surgido alguna dificultad.

Con la misma importancia, quiero agradecer y además dedicar a mi familia la realización de este proyecto. Ellos han sido los encargados de proporcionar todos y cada uno de los medios necesarios para realizar mis estudios, me han brindado toda la ayuda posible, me han animado en los momentos difíciles y han conseguido que únicamente tenga que centrar mis preocupaciones en estudiar.

Answer2Pass – Juego de evaluación interactiva de alumnos

Todas las personas mencionadas, además de muchas otras, han sido en gran medida responsables de hacer un poco más fácil cada año superado en esta etapa universitaria, de la que me llevo gran cantidad de conocimientos, experiencia y que me ha permitido crecer y madurar como persona. La finalización de este proyecto marca el fin de una de las etapas más importantes en mi vida, ahora comienza otra, para la que tengo diversos proyectos y para la cual me siento preparado gracias a la formación que he recibido en mi carrera y el esfuerzo que he realizado.

Índice de contenido

1. Introducción	12
1.1. Motivación del proyecto	12
1.2. Objetivos	14
1.3. Contenido de la memoria	15
2. Análisis	16
2.1. Introducción	16
2.2. Redes sociales en internet	16
2.2.1. Facebook	19
2.2.2. Tuenti	20
2.3. Aplicaciones similares	21
2.3.1. Herramienta 'Quiz' de Moodle	21
2.3.2. Millonaire para Moodle	23
2.3.3. Battery and Coffee Pot Quiz para Moodle	24
2.3.4. Trivial Pursuit	25
2.3.5. Quizazz	26
2.3.6. Testeando	27
2.3.7. Otras aplicaciones similares	30
2.3.8. Comparación de resultados	32
2.4. Análisis de tecnologías y alternativas	33
2.4.1. Plataforma	34
2.4.1.1. Facebook	34
2.4.1.2. Moodle	35
2.4.2. Tecnologías de despliegue	37
2.4.2.1. Google App-Engine	37
2.4.2.2. Amazon EC2	38
2.4.3. Tecnologías de almacenamiento	39
2.4.3.1. JDO	39
2.4.3.2. MySQL	39
2.5. Selección de tecnologías	40
2.6. Diagrama de casos de uso	41

2.7. Requisitos de software	44
2.7.1. Formato seguido en la especificación de requisitos	44
2.7.2. Requisitos funcionales.....	44
2.7.2.1. Requisitos funcionales de operación	45
2.7.2.2. Requisitos funcionales de datos.....	48
2.7.3. Requisitos no funcionales.....	51
2.7.3.1. Requisitos no funcionales de rendimiento	51
2.7.3.2. Requisitos no funcionales de interoperabilidad	51
2.7.3.3. Requisitos no funcionales de interfaz	51
2.7.3.4. Requisitos no funcionales de accesibilidad	52
2.8. Plan de pruebas	52
2.8.1. Formato de las pruebas.....	53
2.8.2. Catálogo de pruebas	54
2.8.3. Matriz Requisitos funcionales – Pruebas	62
3. Diseño	67
3.1. Arquitectura del sistema	67
3.1.1. Tipo de aplicación a desarrollar	67
3.1.2. Modelo a seguir.....	69
3.1.3. Diagrama de componentes	71
3.2 Diseño detallado	73
3.2.1. Diseño de la Interfaz.....	76
3.2.2. Diseño del controlador	91
3.2.3. Diseño del modelo	101
3.3. Diagramas de secuencia	107
3.3.1. Envío de una petición al administrador	108
3.3.2. Aceptar una petición	110
3.3.3. Crear una nueva pregunta.....	112
3.3.4. Responder una pregunta del tablero	114
4. Implementación.....	116
4.1. Aspectos de la implementación	116
4.1.1. Conexión con Facebook	116
4.1.2. JDO	122
4.2. Resultados de las pruebas.....	127

5. Gestión del proyecto	128
5.1. Planificación del proyecto	129
5.1.1. Planificación inicial	129
5.1.2. Planificación final	131
5.2. Medios técnicos empleados	133
5.2.1. Hardware	133
5.2.2. Software	133
5.3. Análisis económico	134
5.3.1. Estimación de costes	135
5.3.2. Costes reales	137
5.4. Análisis de la forma de venta de la aplicación	140
5.4.1. Venta a un cliente	140
5.4.2. Integración de banners publicitarios	141
6. Conclusiones	143
6.1. Conclusiones del proyecto	143
6.2. Conclusiones personales	144
6.3. Líneas futuras	145
ANEXO I - Manual de usuario	147
1. Interfaz de inicio	147
2. Parte dedicada al usuario de tipo alumno	149
2.1. Interfaz inicial de juego	149
2.2. Jugar una nueva partida	150
2.3. Rankings	155
3. Parte dedicada al usuario de tipo profesor	157
3.1. Gestión de preguntas de juego	157
3.2. Gestión de preguntas enviadas por los alumnos	158
3.3. Gestión de peticiones	160
3.4. Gestión de Tableros	161
3.5. Visualización de Rankings y tablero de juego	162
Fuentes de las imágenes utilizadas	163
Bibliografía	163

Índice de figuras

Figura 1: Mapa lógico de ARPANET.....	12
Figura 2: Representación gráfica de red social	17
Figura 3: Millones de usuarios en las redes sociales.....	17
Figura 4: Qué redes sociales prefieren los usuarios.....	18
Figura 5: Crecimiento de la popularidad de Facebook	19
Figura 6: Ejemplo de Quiz , Moodle	22
Figura 7: Captura de la interfaz de 'Millonaire', Moodle	23
Figura 8: Captura de la interfaz 'Coffee Pot', Moodle.....	24
Figura 9: Captura de la Interfaz 'Battery Quiz', Moodle.....	25
Figura 10: Interfaz de juego de 'Trivial Pursuit'.....	26
Figura 11: Captura de 'Quizzaz', Facebook.....	27
Figura 12: Captura de 'Testeando', interfaz de inicio	28
Figura 13: Captura de 'Testeando', interfaz de pregunta	29
Figura 14: Captura de la Interfaz de 'Trivial Pursuit Online'	30
Figura 15: Captura de la Interfaz de 'Quiz Millonario'	31
Figura 16: Captura de la Interfaz de 'Triviador', Facebook	32
Figura 17: Captura de la pantalla de permisos de desarrollador, Facebook.....	34
Figura 18: Interfaz de administrador, Google App Engine	38
Figura 19: Diagrama de casos de uso	42
Figura 20: Conexión Cliente - Facebook – Google App Engine	68
Figura 21: Modelo Vista Controlador.....	70
Figura 22: Componentes de la aplicación	72
Figura 23: Relaciones existentes entre las clases de la aplicación.....	74
Figura 24: Interfaz de inicio, Answer2Pass.....	77
Figura 25: Interfaz de juego I, Answer2Pass	78
Figura 26: Interfaz de juego II, Answer2Pass	79
Figura 27: Interfaz de juego III, Answer2Pass	79
Figura 28: Interfaz de juego IV, Answer2Pass.....	80
Figura 29: Interfaz de administración I, Answer2Pass	83
Figura 30: Interfaz de administración II, Answer2Pass	84

Answer2Pass – Juego de evaluación interactiva de alumnos

Figura 31: Interfaz de administración III, Answer2Pass	85
Figura 32: Interfaz de administración IV, Answer2Pass	85
Figura 33: Interfaz de administración V, Answer2Pass	86
Figura 34: Interfaz de administración VI, Answer2Pass	87
Figura 35: Interfaz de Rankings, Answer2Pass.....	88
Figura 36: Interfaz de visualización de nuevos tableros, Answer2Pass	89
Figura 37: Envío de petición	108
Figura 38: Aceptar petición	110
Figura 39: Crear nueva pregunta.....	112
Figura 40: Responder una pregunta.....	114
Figura 41: Proceso de acceso a la aplicación	117
Figura 42: Autenticación con Facebook	118
Figura 43: Obtención de librería Javascript.....	118
Figura 44: Comprobar estado de autenticación.....	119
Figura 45: Obtención de permisos	120
Figura 46: Envío de notificaciones.....	121
Figura 47: Publicar puntuación en el muro	122
Figura 48: Definición de clases JDO I.....	123
Figura 49: Definición de clases JDO II.....	124
Figura 50: Definición de clases JDO III.....	125
Figura 51: JDO, creación de objetos.....	126
Figura 52: JDO, Consulta por clave primaria	126
Figura 53: JDO, consulta y eliminación de un grupo de objetos	127
Figura 54: Planificación inicial, diagrama de Gantt	130
Figura 55: Planificación final, diagrama de Gantt	132
Figura 56: Manual de usuario, interfaz inicial.....	148
Figura 57: Manual de usuario, interfaz principal de juego	150
Figura 58: Manual de usuario, interfaz de respuesta a pregunta	151
Figura 59: Manual de usuario, interfaz de creación de pregunta.....	153
Figura 60: Manual de usuario, interfaz de preguntas recibidas por otros alumnos.....	155
Figura 61: Manual de usuario, mensaje publicado en el muro de Facebook	155
Figura 62: Manual de usuario, interfaz de rankings.....	156
Figura 63: Manual de usuario, identificador de vídeo	157

Answer2Pass – Juego de evaluación interactiva de alumnos

Figura 64: Manual de usuario, interfaz de gestión de preguntas de juego	158
Figura 65: Manual de usuario, interfaz de gestión de preguntas de alumnos	159
Figura 66: Manual de usuario, interfaz de gestión de peticiones.....	160
Figura 67: Manual de usuario, interfaz de gestión de tableros I	161
Figura 68: Manual de usuario, interfaz de gestión de tableros II	162

Índice de tablas

Tabla 1: Comparación de resultados según los objetivos	33
Tabla 2: Formato de la tabla de requisitos funcionales	44
Tabla 3: Formato de la tabla de pruebas	53
Tabla 4: Matriz Requisitos - Pruebas.....	66
Tabla 5: Formato de la tabla de clases	91
Tabla 6: Clase SendRequestServlet	92
Tabla 7: Clase FieldServlet.....	93
Tabla 8: Clase LoadQuestionServlet.....	94
Tabla 9: Clase ScoreboardServlet	95
Tabla 10: Clase CheckQuestionsServlet	95
Tabla 11: Clase QuestionsServlet	96
Tabla 12: Clase ManageFieldServlet	97
Tabla 13: Clase GenerateFieldServlet	98
Tabla 14: Clase ManageRequestServlet	98
Tabla 15: Clase UserServlet.....	99
Tabla 16: Clase IncreaseAnswerPointsServlet.....	99
Tabla 17: Clase CheckAnswersServlet	101
Tabla 18: Clase Petitions	102
Tabla 19: Clase User	103
Tabla 20: Clase TrueFalse	104
Tabla 21: Clase OneAnswer.....	105
Tabla 22: Clase MultiAnswer.....	106
Tabla 23: Clase Field.....	106

Answer2Pass – Juego de evaluación interactiva de alumnos

Tabla 24: Resultados de las pruebas	128
Tabla 25: Herramienta hardware I	133
Tabla 26: Herramienta hardware II	133
Tabla 27: Herramientas software	134
Tabla 28: Coste de personal estimado I	135
Tabla 29: Coste de personal estimado II	135
Tabla 30: Coste de hardware estimado	136
Tabla 31: Coste de software estimado	136
Tabla 32: Costes indirectos estimados	136
Tabla 33: Costes totales estimados	137
Tabla 34: Costes reales de personal I	137
Tabla 35: Costes reales de personal II	138
Tabla 36: Costes reales de hardware	138
Tabla 37: Costes reales de software	138
Tabla 38: Costes reales indirectos	139
Tabla 39: Costes reales totales	139
Tabla 40: Integración de dos banners publicitarios	141
Tabla 41: Integración de cuatro banners publicitarios	142

Índice de requisitos

Requisitos 1: Requisitos funcionales de operación	49
Requisitos 2: Requisitos funcionales de datos	51
Requisitos 3: Requisitos no funcionales de rendimiento	52
Requisitos 4: Requisitos no funcionales de interoperabilidad	52
Requisitos 5: Requisitos no funcionales de interfaz	53
Requisitos 6: Requisitos no funcionales de accesibilidad	53

Índice de pruebas

Pruebas 1: Catálogo de pruebas	63
--------------------------------------	----

1. Introducción

1.1. Motivación del proyecto

Se dice que internet es la ‘Autopista de la Información’. Esto es debido a que, al ser una red formada por redes de ordenadores de alcance mundial, millones de personas de todo el mundo tienen la posibilidad de acceder de forma sencilla a una cantidad extensa y variada de información. Según estadísticas obtenidas ¹en 2011, más de 2.000 millones de usuarios de todo el mundo ya usan Internet, lo que supone el 30% de la población mundial actual. Estas cifras dan para pensar la cantidad de información diaria intercambiada por los usuarios de dicha ‘autopista’.

Internet, conocida por otros como la ‘Red de Redes’, remonta su origen al año 1960 con ARPANET (*Advanced Research Projects Agency Network*), red creada por una agencia del Departamento de Defensa de Estados Unidos llamada DARPA (*Defense Advanced Research Project Agency*). Dicha agencia, llamada ARPA anteriormente, tenía como principal objetivo conseguir enlazar diferentes ordenadores de la época para mejorar la potencia de procesamiento de los mismos y descentralizar así el almacenamiento de información.

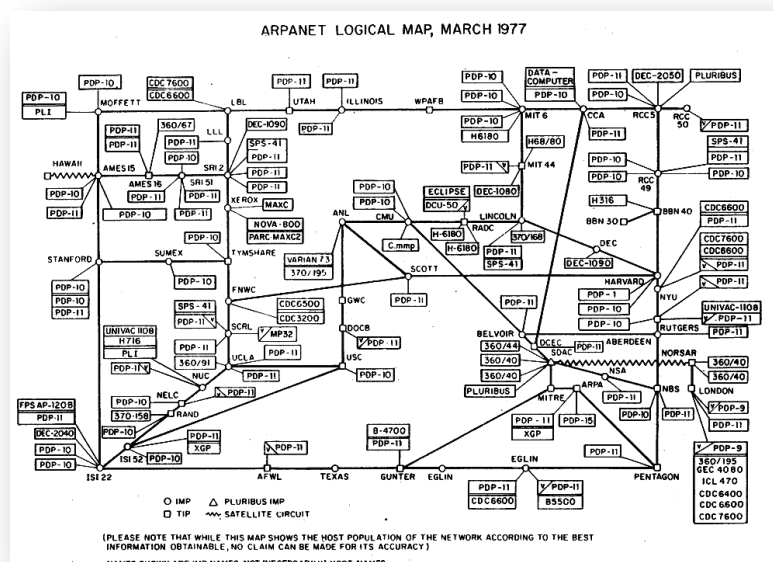


Figura 1: Mapa lógico de ARPANET ¹²

¹ Datos obtenidos de [21]

² Imagen obtenida de [1]

Posteriormente, partiendo de la idea de red de comunicaciones de ARPANET, se fueron desarrollando nuevas redes y protocolos hasta llegar a 1990, año en el que gracias a la introducción de nuevas facilidades de interconexión y herramientas se inició el auge de Internet. Este aumento masivo de usuarios de la red fue confirmado en 2006, año en el que se alcanzó la cifra de 1.100 millones de usuarios.

El desarrollo de Internet y la evolución de los servicios que ofrece a los usuarios ha supuesto un gran impacto en el campo de la educación. Estudiantes y profesores pueden acceder a cantidad de información procedente de todo el mundo que les permita ampliar y fijar sus conocimientos, permitiendo el acceso a contenido multimedia, textual y herramientas de trabajo costosas de encontrar fuera de la red. De la misma forma, Internet proporciona una gran cantidad de recursos que facilitan en gran medida la interacción profesor – alumno y alumno - alumno (Servicios como el correo electrónico, herramientas como ‘Moodle’ descrita más adelante), facilitando el intercambio de información e incrementando así la facilidad de aprendizaje. Por otro lado, estos recursos también han sido útiles para ayudar a los alumnos a aprender de una forma diferente a las clases convencionales, por medio de exámenes en línea, aplicaciones y juegos multimedia etc. Por último, cabe destacar que gracias a dichos recursos han surgido ideas que suponen un gran avance en el campo de la enseñanza, como pueden ser la impartición de clases y másters y carreras universitarias a distancia.

Otro de los servicios que ofrece internet, y que se encuentra muy de moda actualmente, son las redes sociales. Se trata de páginas web encaminadas al establecimiento de relaciones entre personas, cuyo fin es formar una red en la que todas estén conectadas por diversos tipos de enlaces como puede ser la amistad, el parentesco, intereses comunes etc. Las redes sociales permiten entre otras cosas compartir fotografías, estados, comentarios, intereses, opiniones, vídeos etc. Este tipo de servicio web también puede suponer un motor empleado para motivar a los usuarios a hacer uso de aplicaciones con fines educativos, que serían accesibles a un mayor número de personas y que brindarían la posibilidad de ser compartidas en todo momento. Estas redes sociales ya cuentan con gran cantidad de aplicaciones y juegos que permiten la colaboración entre usuarios para alcanzar mejores resultados, los cuales podrán ser compartidos con todos los amigos de la red social.

La motivación de este proyecto reside en el desarrollo de una herramienta más, distinta de las ya implementadas, que suponga un nuevo aporte a dicho avance de la educación en medios como Internet. Se pretende implementar una nueva idea útil e innovadora que facilite el proceso de aprendizaje y evaluación de los alumnos, permitiendo la adquisición y fijación de conocimientos lejos de las a menudo pesadas clases teóricas y exámenes. Con todo ello se busca una aplicación que suponga un nuevo recurso tanto para profesores, ya que está pensado para evaluar los conocimientos aprendidos en clase, como para los alumnos, que pueden utilizar para aprender o fijar los conocimientos aprendidos en clase.

En el siguiente apartado se encuentran definidos más a fondo todos los objetivos del proyecto a desarrollar.

1.2. Objetivos

El principal objetivo perseguido por este proyecto es **el desarrollo de una aplicación web denominada Answer2Pass que permita a los profesores evaluar a los alumnos de una forma diferente, consiguiendo que estos aprendan jugando y colaborando unos con otros. Dicha aplicación debe permitir registrar y compartir todas las puntuaciones obtenidas, alcanzando así un cierto grado de competitividad que la pueda hacer un poco más adictiva.** La aplicación debe permitir a los alumnos fijar los conocimientos impartidos en clase por medio de la respuesta a preguntas, elaboradas por los profesores y los propios alumnos, dispuestas en un tablero de juego. Los objetivos específicos de dicho proyecto se encuentran enumerados a continuación:

- La aplicación a desarrollar debe permitir a los alumnos registrados **jugar un número indefinido de partidas respondiendo las preguntas propuestas en cada una de ellas.** Dichas preguntas deben sucederse según el avance en un tablero de juego, que debe estar formado por casillas de diferente color según el tema de la pregunta.
- Los usuarios podrán hacer uso de comodines, entre los que se encuentra la obligación de **crear una nueva pregunta** por parte del usuario con la finalidad de ser enviada al tablero de otro alumno. **Esto último es llevado a cabo para fomentar la interacción entre alumnos perseguida la aplicación.**
- Todas las puntuaciones obtenidas deben almacenarse y disponerse en forma de rankings, cuyo objetivo es **aumentar el grado de competitividad** y por lo tanto, la frecuencia de juego por parte de los alumnos.
- En lo que a los profesores se refiere, la aplicación tiene como objetivo la **participación de los mismos en la confección de la pila de preguntas propuestas a los alumnos en el juego.** Así como en la **moderación y calificación** de todas las preguntas enviadas entre ellos. El profesor también podrá gestionar las peticiones de acceso recibidas y los tableros de juego.
- Otro de los objetivos de la aplicación es su desarrollo íntegro en inglés, para ayudar de esta manera a **fomentar el idioma entre los alumnos.**

1.3. Contenido de la memoria

La presente memoria se encuentra estructurada de la siguiente forma:

- **Capítulo 1, Introducción:** En este primer capítulo se introduce al lector en el contexto de realización de dicho proyecto, indicando la motivación por la que se ha llevado a cabo y especificando los objetivos perseguidos.
- **Capítulo 2, Análisis:** En dicho apartado se procede a realizar el análisis de recursos web como las redes sociales, tecnologías y alternativas candidatas para el despliegue, y aplicaciones similares implementadas, estableciendo en qué medida satisfacen o no los objetivos perseguidos. En este mismo apartado también se especifican los casos de uso, los requisitos y el plan de pruebas propio de la aplicación.
- **Capítulo 3, Diseño:** En este capítulo enfocado al diseño de la aplicación, se establece la arquitectura del sistema identificando el tipo de aplicación a desarrollar, el modelo seguido y el diseño de los componentes que la forman.
- **Capítulo 4, Implementación:** Este apartado, relacionado con la implementación del sistema, aborda a nivel de código aspectos y técnicas relevantes para implementar la aplicación. De la misma forma, se encarga de enunciar el resultado obtenido tras la realización del plan de pruebas.
- **Capítulo 5, Gestión del Proyecto:** Otro de los puntos a desarrollar en la presente memoria es el dedicado a la gestión del proyecto, cuyo objetivo es analizar la planificación inicial y final del proyecto y el coste económico que ha supuesto. De la misma forma, se analizan diferentes opciones candidatas como posible vía de venta del proyecto.
- **Capítulo 6, Conclusiones y Líneas futuras:** Se dispone el apartado que enuncia la conclusión personal y del proyecto, así como las posibles mejoras que podrían ser implementadas en la aplicación.
- **Anexo I, Manual de usuario:** Este apartado del anexo es el encargado de albergar el manual que debe seguir un usuario para hacer un correcto uso de la aplicación. En dicho punto se especifican, con ayuda de imágenes detalladas, todas las funcionalidades de la aplicación y los pasos necesarios para llevarlas a cabo.
- **Referencias:** Apartado final de la memoria donde se incluyen todas las fuentes de información consultadas.

2. Análisis

2.1. Introducción

A continuación se tratará de realizar un análisis completo sobre la aplicación a desarrollar, obteniendo así una especificación detallada del sistema que pueda servir de base para el diseño del mismo.

El análisis se realizará sobre los siguientes apartados:

- **Redes sociales:** Se explicarán brevemente redes sociales como Facebook y Tuenti, explicando las diferencias entre ellas, los efectos que tienen sobre la sociedad hoy en día y las ventajas que supone el desarrollo de una aplicación como Answer2Pass sobre ellas.
- **Aplicaciones similares:** Se analizarán aplicaciones parecidas, diferenciándolas de la aplicación a desarrollar en este proyecto e identificando que carencias tienen en cuanto al objetivo perseguido por Answer2Pass.
- **Análisis de tecnologías y alternativas:** Se analizarán todas las tecnologías pensadas para desarrollar la aplicación, tanto las elegidas finalmente como las posibles alternativas.
- **Elección final:** Se proporcionarán todas las razones que explican que tecnologías se han usado finalmente para la realización de la aplicación, indicando todas las ventajas e inconvenientes que supone dicha elección.
- **Diagrama de casos de uso:** Se incluirán diagramas que representan el comportamiento del sistema ante la acción distintos actores, en este caso Administrador y Usuario.
- **Requisitos Software:** En este apartado se analizarán todos los requisitos necesarios para dar una descripción completa de la aplicación a desarrollar.
- **Plan de Pruebas:** Finalmente, se establecerá un plan de pruebas que sea capaz de verificar la funcionalidad de los requisitos de la aplicación.

2.2. Redes sociales en internet

Las redes sociales son sitios web que forman una estructura compuesta por un gran número de personas que participan en ella. A través de ellas, se utilizan los vínculos existentes entre los usuarios para enlazar unos con otros y formar así una red en la que todo el mundo puede estar comunicado. De esta forma se pueden compartir actividades, comentarios, ideas, vídeos, fotografías etc.



Figura 2: Representación gráfica de red social³

Las redes sociales son aplicaciones de moda en la actualidad, hasta tal punto que ya es difícil encontrar alguien que no pertenezca a alguna de ellas. Por ello, estas crecen a una velocidad asombrosa, alcanzando cifras que se muestran en el siguiente gráfico⁴:

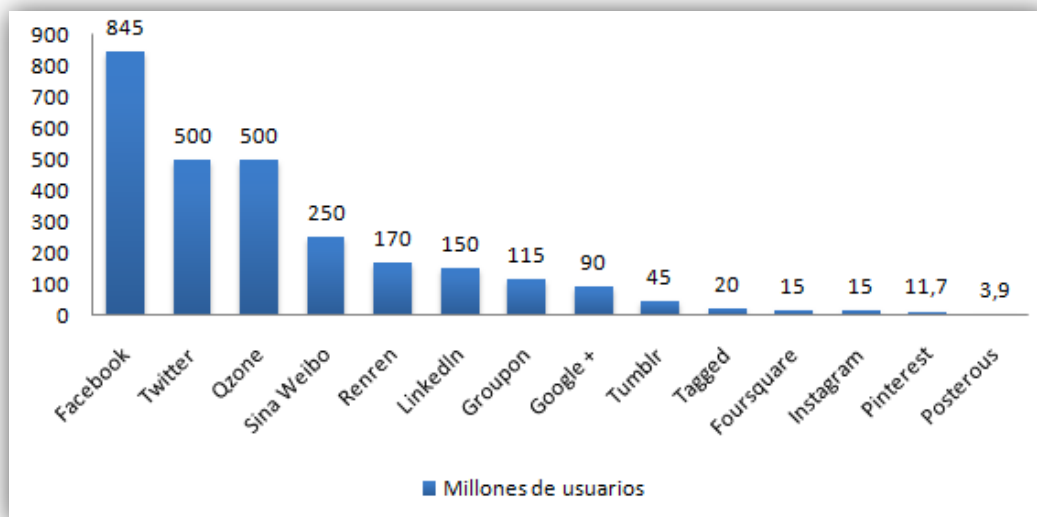


Figura 3: Millones de usuarios en las redes sociales

³ Imagen obtenida de [II]

⁴ Datos obtenidos de [27]

Como podemos observar en el gráfico anterior, las redes sociales que cuentan con más usuarios en el mundo son Facebook y Twitter, alcanzando cifras de 845 y 500 millones de usuarios. De cerca les siguen las redes sociales asiáticas Qzone, Sina Weibo y Renren y posteriormente LinkedIn, Groupon y Google Plus.

Por otro lado, en cuanto a España se refiere, si nos fijamos en las estadísticas obtenidas a través de un estudio realizado por IAB Spain (*“Interactive Advertising Bureau”*), podemos ver que Tuenti y Youtube se convierten después de Facebook y con mucha diferencia de las otras redes sociales, en las preferidas por los usuarios.

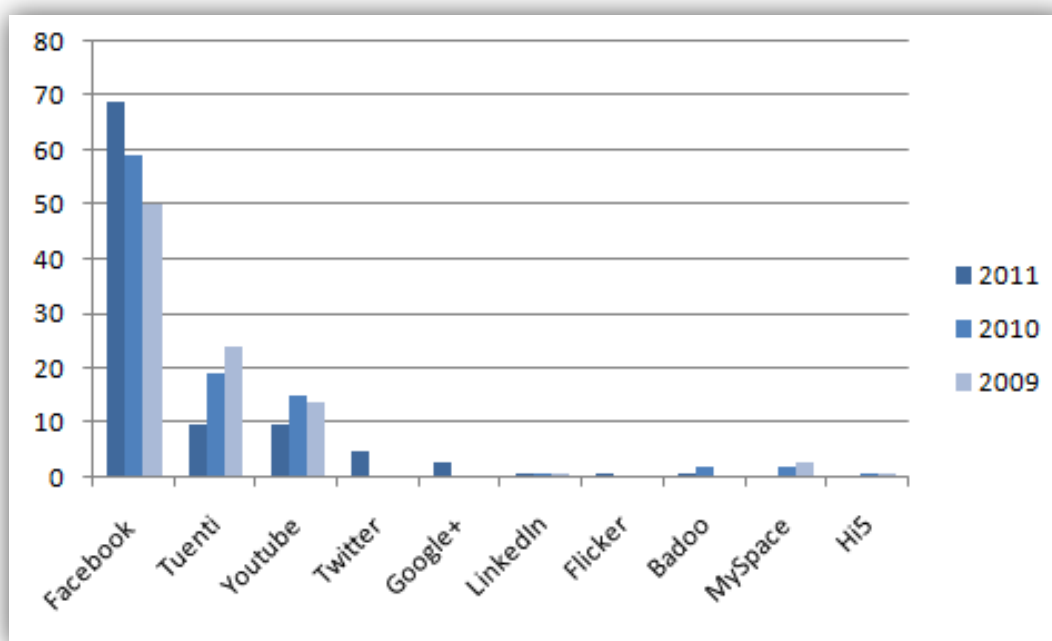


Figura 4: Qué redes sociales prefieren los usuarios⁵

El gráfico muestra el porcentaje de usuarios encuestados por año que prefieren las redes sociales planteadas. El número de usuarios que IAB Spain encuestó fue:

- 725 usuarios en 2011
- 548 usuarios en 2010
- 503 usuarios en 2009

Por todos estos datos mostrados anteriormente se ha decidido analizar un poco más detalladamente las redes sociales Facebook y Tuenti como principales candidatas a poder albergar aplicaciones que persigan el mismo objetivo que la desarrollada en este proyecto.

⁵ Datos obtenidos de [10]

2.2.1. Facebook

Facebook es una red social creada por Mark Zuckerberg y fundada en 2004. Originalmente fue implementada únicamente para estudiantes pertenecientes a la Universidad de Harvard, pero en la actualidad se encuentra disponible para cualquiera que disponga de una cuenta de correo electrónico. Actualmente se encuentra en lo más alto de la lista de redes sociales, contando con más de 900 millones de usuarios y traducciones a 70 idiomas.

El gráfico siguiente muestra el nivel de popularidad (en millones de usuarios) alcanzado por Facebook desde su fundación en 2004:

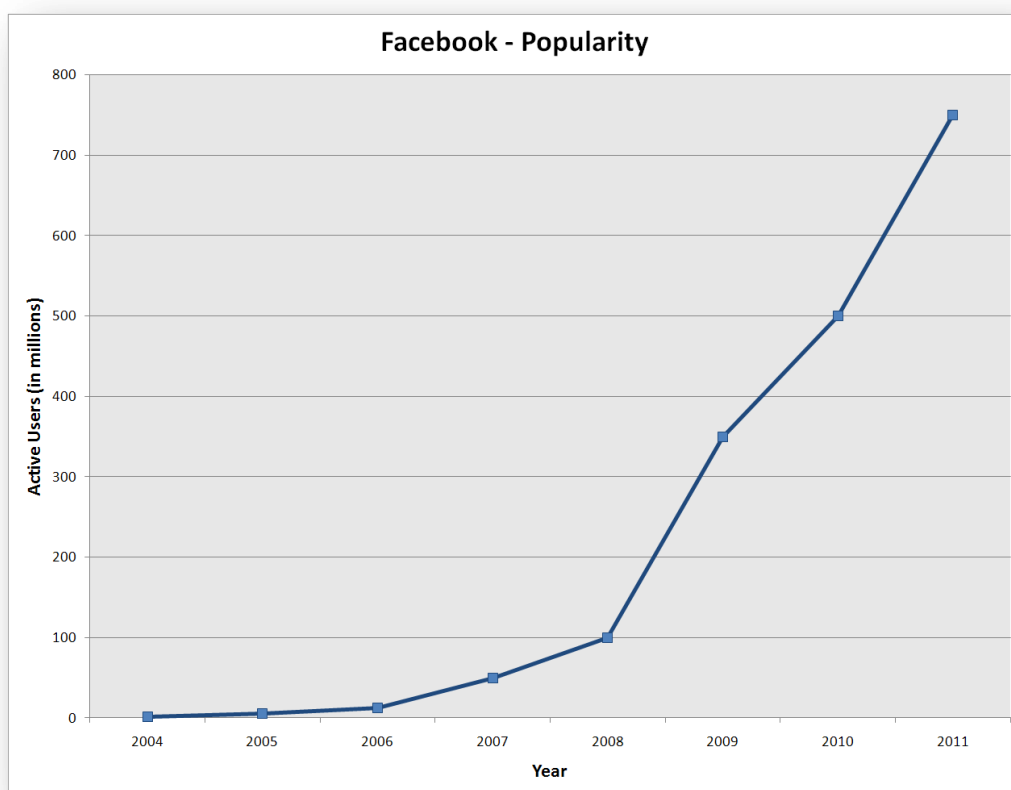


Figura 5: Crecimiento de la popularidad de Facebook⁶

Esta red social permite al usuario la creación de un perfil formado entre muchas otras cosas por información básica como el cumpleaños o el sexo, lugares de estudio y trabajo, relaciones personales y familiares, deporte, música, libros y películas preferidas etc.

⁶ Imagen obtenida de [III]

En cuanto los servicios que ofrece se pueden destacar el intercambio de mensajes a través del ‘muro’ o mensajería privada, y la compartición de fotografías. Este ‘muro’ es un espacio reservado en la red social destinado a la compartición de actividades y de mensajes recibidos por miembros de la lista de amigos. En cuanto a las fotografías, Facebook cuenta ya con 5.000 millones de fotos de usuario, que se pueden compartir, comentar y ‘gustar’. Cabe destacar que en noviembre de 2011 el creador de Facebook Mark Zuckerberg anunció la ‘Biografía’, cuyo fin es sustituir al ‘muro’ agilizando el paseo de los usuarios por los perfiles de sus contactos.

Por otro lado, Facebook cuenta con gran variedad de juegos y aplicaciones que permiten al usuario compartir acciones y puntuaciones obtenidas con la red. Dichas aplicaciones pueden ser implementadas por cualquier usuario con cuenta verificada (A través del teléfono móvil, envío del DNI etc.), siendo esta funcionalidad la más significativa en lo que al desarrollo del proyecto se refiere.

También, Facebook cuenta con una utilidad interesante relacionada con la creación de grupos de interés y páginas. Los grupos son secciones de la red social creadas por los usuarios para poder proponer convocatorias o discusiones de un tipo determinado, y las páginas son secciones encaminadas a compartir marcas o personajes específicos.

2.2.2. Tuenti

Tuenti es una red social española inaugurada en 2006 que fue creada por Félix Ruiz, Joaquín Ayuso, Kenny Bentley y Zaryn Dentzel. Al igual que Facebook, al principio sólo iba dirigida a universitarios, y más tarde debido a su éxito permitió la entrada a todos aquellos que contaran con cuenta de correo electrónico.

Se trata de una red social enfocada principalmente a usuarios españoles, aunque Telefónica (accionista mayoritario), tiene la intención de expandirla a otros países como los europeos y latinoamericanos a medio-largo plazo.

Una vez registrados, los usuarios de la red social pueden crear un perfil con información básica como el sexo, la edad o la procedencia, y otros datos como los intereses, lugares favoritos o páginas seguidas. Los usuarios pueden escribir comentarios a sus amigos en el ‘tablón’ destinado para ello o por medio de mensajes privados. Por otro lado, también tienen la posibilidad de subir fotografías, escribir entradas y compartir estados.

Tuenti también ofrece la posibilidad de crear nuevas páginas y ‘eventos’ de temas determinados. Estos últimos persiguen la finalidad de debatir sobre un tema en concreto o proponer convocatorias a los que los usuarios indiquen su asistencia. Al igual que Facebook, Tuenti cuenta con un amplio abanico de juegos que hacen pasar

un buen rato al usuario y que permiten competir con otros amigos a través de la compartición de puntuaciones.

Esta red social cuenta, desde el 21 de diciembre de 2010, con la prestación de servicios de telefonía móvil como OMV (*Operador Móvil Virtual*) de Tuenti, ofreciendo *Smartphones* y tarifas especiales a los usuarios que la posean. Este innovador tipo de servicio puede marcar la diferencia entre las demás redes sociales.

Cabe destacar que a diferencia de Facebook, Tuenti no posee todavía entorno dedicado a desarrolladores, por lo que no es posible la implantación de aplicaciones en la red social.

2.3. Aplicaciones similares

A continuación se van a analizar diferentes aplicaciones ya implementadas parecidas al software desarrollado en este proyecto. Se explicará su desarrollo, objetivos y características y se elaborará una lista de requisitos que servirá para comparar las ventajas e inconvenientes de dichas aplicaciones y Answer2Pass.

2.3.1. Herramienta ‘Quiz’ de Moodle

En primer lugar se va a realizar una toma de contacto con **Moodle**, explicando qué es, qué objetivos persigue y qué características tiene.

A grandes rasgos Moodle es un paquete software especializado en proporcionar una serie de herramientas que facilitan la creación y gestión de aplicaciones web educativas.

Moodle, procedente del acrónimo Modular *Object-Oriented Dynamic Learning Environment*, es un paquete software utilizado para facilitar la creación y el soporte de aplicaciones educativas en la red, de esta forma es conocido por ser un CMS, LMS o VLE (*Course Management System*, *Learning Management System* y *Virtual Learning Environment* respectivamente).

Una de las herramientas que Moodle posee es la de tipo **Quiz**. Este módulo permite a los profesores diseñar cuestionarios incluyendo preguntas de todo tipo, como por ejemplo Verdadero y Falso, una sola respuesta o de respuesta múltiple. Todas las preguntas creadas se guardarán en un banco de preguntas para poder ser usadas de nuevo en otros cuestionarios.

Answer2Pass – Juego de evaluación interactiva de alumnos

Los cuestionarios pueden realizarse varias veces, registrando la calificación en cada una de las ocasiones. El profesor/a puede configurarlo de tal forma que cambie el tipo de mensaje a mostrar, muestre las respuestas correctas al final, seleccionar preguntas aleatorias de diferentes categorías obtenidas a partir de las ya creadas, cambiar el tipo de puntuación, el número de preguntas por página etc.

Crear un cuestionario con esta herramienta se convierte en una tarea sencilla para los profesores. En resumen estos tienen que seguir los siguientes pasos:

- 1- Ir a la página principal del curso y hacer click en 'Cuestionario', encontrado en la parte de 'Agregar Actividad'.
- 2- Rellenar un formulario con el nombre, descripción, fechas y la calificación del cuestionario.
- 3- Seleccionar en el desplegable de la columna derecha la categoría de las preguntas a incluir en el diccionario.
- 4- Crear nuevas preguntas o usar las ya almacenadas.
- 5- Agregar el cuestionario.

Un ejemplo de cuestionario creado usando esta herramienta sería:



Figura 6: Ejemplo de Quiz, Moodle⁷

⁷ Imagen obtenida de [IV]

2.3.2. Millionaire para Moodle

Millonaire es un juego educativo implementado en Moodle que pertenece al módulo llamado Game-Module. Es un juego parecido al concurso televisivo ‘¿Quién quiere ser millonario?’, que permite ir aumentando la puntuación final respondiendo una serie de preguntas planteadas haciendo uso de diferentes tipos de comodines.

Las reglas de este juego son simples, el usuario tiene que ir respondiendo todas las preguntas que se le muestran por pantalla, acertando para poder continuar con la siguiente. Si falla cualquier pregunta el juego acabará. El acierto o fallo de la pregunta podrá visualizarse en la parte de abajo a la derecha de la pantalla.

Se podrán usar comodines para facilitar llegar a la respuesta correcta en las preguntas. Estos tienen la siguiente funcionalidad:

- **50:50:** Con este comodín el usuario elimina dos respuestas posibles, pasando a tener el 50% de probabilidad para acertar la pregunta.
- **Comodín de la llamada:** El usuario recibirá una respuesta sugerida simulando haber realizando dicha llamada.
- **Comodín del público:** El usuario recibirá la simulación del porcentaje de personas del público que ha seleccionado cada respuesta.

La interfaz de juego es la siguiente:

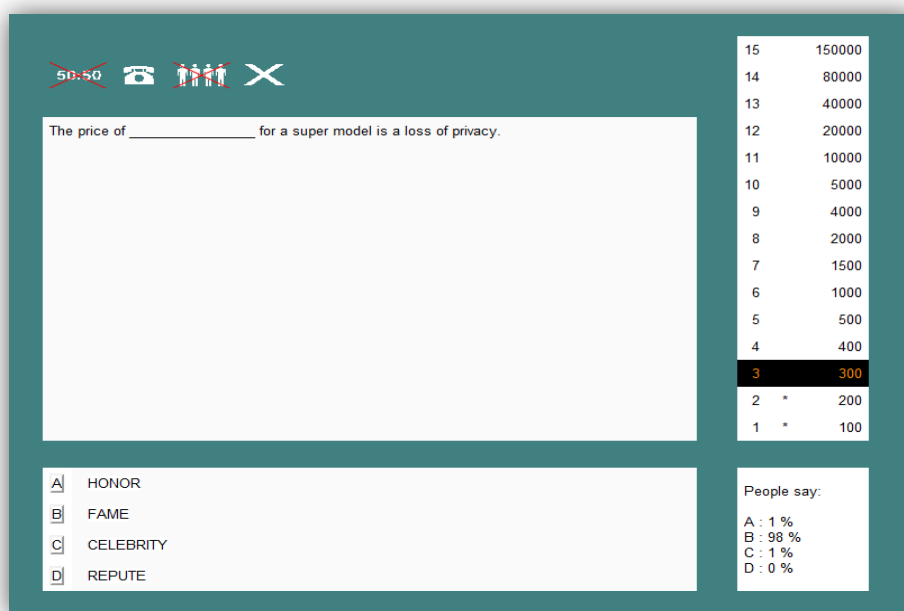


Figura 7: Captura de la interfaz de 'Millonaire', Moodle

Como se observa en la interfaz anterior, la pantalla queda dividida en 5 partes diferenciadas. Empezando por la esquina superior izquierda, se encuentra la parte dedicada a los comodines del juego, que están representados por iconos descriptivos y tachados con un aspa roja si ya han sido usados. Por otro lado, bajo la parte dedicada a los comodines, se encuentran las regiones donde los usuarios pueden visualizar el enunciado de las preguntas y las respuestas a las mismas. Finalmente cabe destacar las zonas situadas a la derecha de la pantalla, que tienen la función de mostrar la puntuación obtenida por cada pregunta (Parte superior) y de mostrar los mensajes del sistema (Parte inferior).

2.3.3. Battery and Coffee Pot Quiz para Moodle

Estos dos juegos parecidos a Answer2Pass son un tipo de 'Quiz-Game' implementado en Flash para Moodle, lo que le da una apariencia bastante atractiva.

El juego básicamente se basa en responder las preguntas mostradas por pantalla. El acierto de estas se verá representado aumentando la carga de una pila o el café en la jarra respectivamente. La puntuación final vendrá determinada por la cantidad de batería y café que se haya conseguido.

Las interfaces de juego son las siguientes:

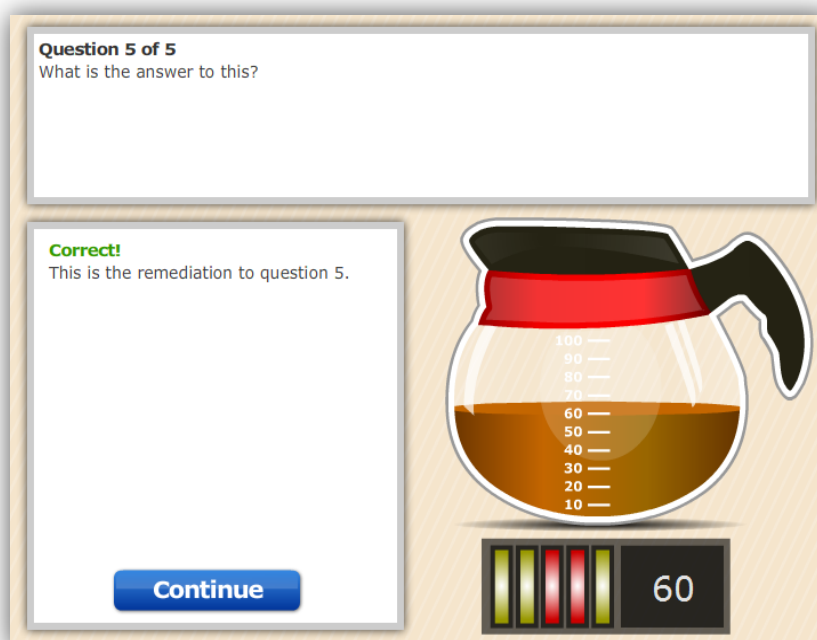


Figura 8: Captura de la interfaz 'Coffee Pot', Moodle

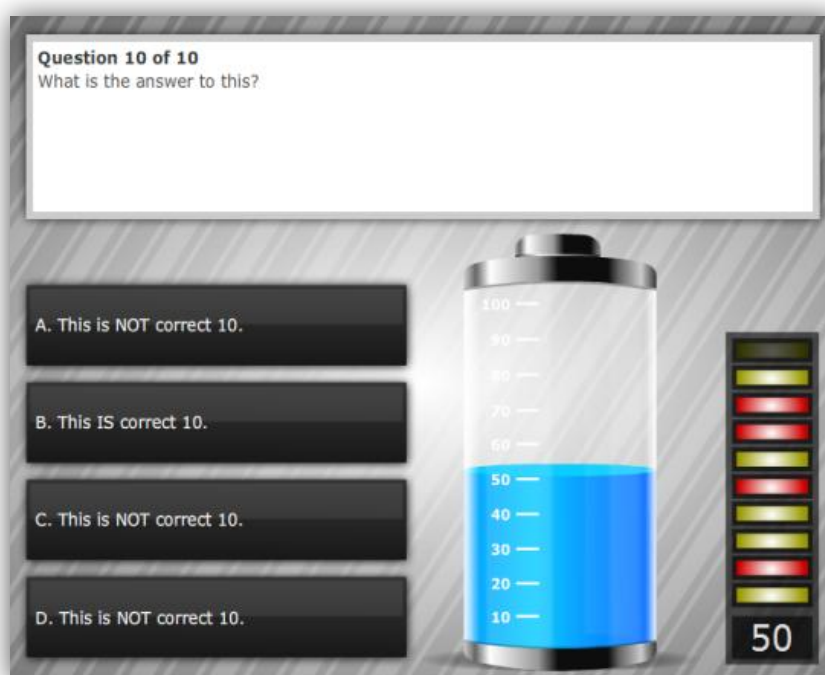


Figura 9: Captura de la Interfaz 'Battery Quiz', Moodle

Las dos interfaces anteriores poseen un diseño muy parecido, ambas cuentan con una zona donde se muestra el enunciado de la pregunta y las respuestas posibles; que representa la puntuación obtenida en la partida; y finalmente, otra región donde se muestra un objeto (batería o taza de café) que indica el porcentaje de partida completado.

2.3.4. Trivial Pursuit

El Trivial Pursuit es un juego de mesa inventado por Scott Abbott y Chris Haney en diciembre de 1979 y lanzado al mercado en 1981. Se comenzaron vendiendo 20 millones de copias tres años después de su lanzamiento y, hasta 2004 después de que Hasbro compra los derechos, se han vendido cerca de 88 millones de ejemplares en 26 países y 17 idiomas.

Este famoso juego de mesa se basa en ir respondiendo preguntas completando los radios de una rueda con los 'quesitos' obtenidos como premio al conseguir acertar cada una de las seis preguntas especiales dispuestas en el tablero.

El trivial ha pasado de ser un juego de mesa a estar desarrollado para una gran cantidad de plataformas. Lo podemos encontrar como juego de PC, de múltiples videoconsolas, tablets o dispositivos móviles. En concreto se va a analizar el juego implementado para Ipad, que quizá sea el que más se asemeje a este proyecto en cuestión.

La interfaz correspondiente a las dos modalidades de juego es la siguiente:



Figura 10: Interfaz de juego de 'Trivial Pursuit'⁸

Como se puede observar, aparte del modo clásico se ha añadido el modo 'Pursuit', que se basa en un tablero con preguntas de distinto tema generadas aleatoriamente que el usuario debe responder para avanzar en el tablero y llegar lo antes posible al final. Este modo cuenta con comodines que pueden beneficiar al usuario doblando el número de su tirada o transportarlo a otra zona del tablero. Dichos comodines son muy similares a los que se van a implementar en la aplicación, tendrán funciones como la elección de comodines aleatorios, temas escogidos al azar, incremento del tiempo de respuesta, aumento en el número de tirada y el que quizá es más significativo en cuanto a los objetivos del juego, la funcionalidad de crear nuevas preguntas para ser respondidas por otros usuarios.

Para concluir el análisis de esta aplicación, cabe destacar que otra de las mejoras más significativas de esta nueva versión de Trivial para Ipad es el modo multijugador, donde se puede jugar online vía WIFI.

2.3.5. Quizazz

Quizazz es una aplicación Web implementada en Facebook que permite crear y participar en cuestionarios creados por otros usuarios. De esta forma los participantes podrán responder las preguntas de cuestionarios de múltiples temas y podrán compartir su puntuación final visualizando las respuestas dadas por todos los usuarios.

⁸ Imagen obtenida de [V]

Answer2Pass – Juego de evaluación interactiva de alumnos

La aplicación permite entre otras cosas buscar cuestionario, por nombre, por más popular y por nuevos; crear nuevo cuestionario, seleccionando el tipo e incluyendo las preguntas con sus imágenes y respuestas; y ver y editar el perfil de usuario.

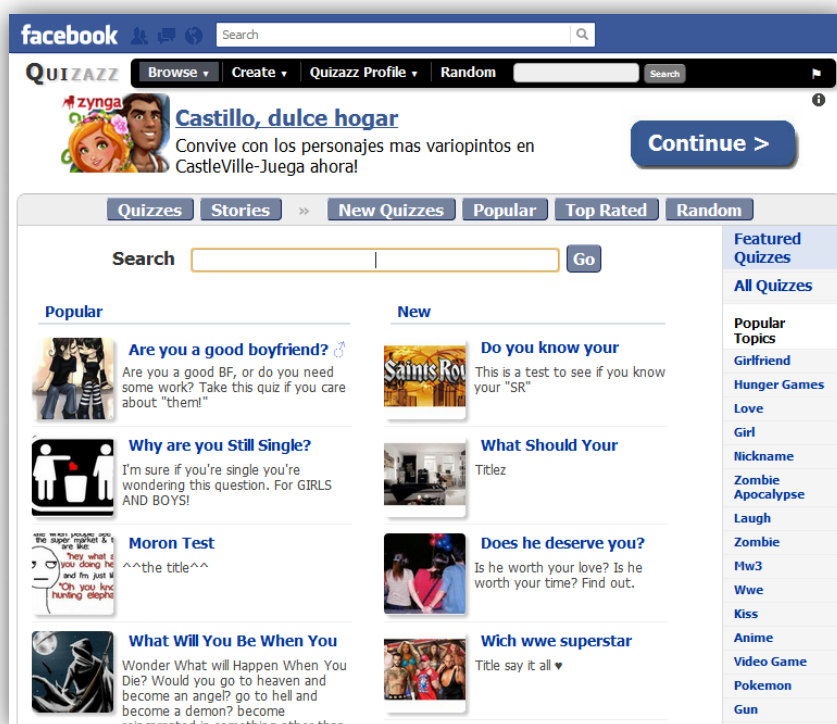


Figura 11: Captura de 'Quizzaz', Facebook

2.3.6. Testeando

Testeando⁹ es una herramienta web educativa pensada para profesores y alumnos españoles y norteamericanos. Es una aplicación que consiste en un juego de preguntas y respuestas de tipo test o trivial, cuyo principal objetivo es conseguir que los alumnos aprendan y complementen los conceptos enseñados en el colegio o instituto de una forma más divertida. Actualmente, Testeando cuenta con una base de datos formada por 1132 test de 64 asignaturas diferentes y unas 23.351 preguntas (aproximadamente 1000 son añadidas por mes).

⁹ Alojada en la dirección web <http://www.testeando.es>



Figura 12: Captura de 'Testeando', interfaz de inicio

En cada partida de juego, los alumnos tienen que responder 10 preguntas del tema y curso elegido, cada una de ellas con cuatro preguntas diferentes. Todas las preguntas son de una sola respuesta, cuyo acierto supone un aumento en la puntuación en proporción al tiempo empleado en responder. Por otro lado, las preguntas falladas contarán de forma negativa. Los alumnos pueden ayudarse de comodines para responder las preguntas, estos serán del tipo 50% (disminuyendo a 2 el número de respuestas posibles), 2x1 (permitiendo marcar 2 de las 4 respuestas posibles) o 'La clase' (mostrando cuál de las respuestas es la más votada). Para repasar y fijar los conceptos, los alumnos pueden repetir los test todas las veces que quieran. La siguiente imagen muestra la pantalla de juego:

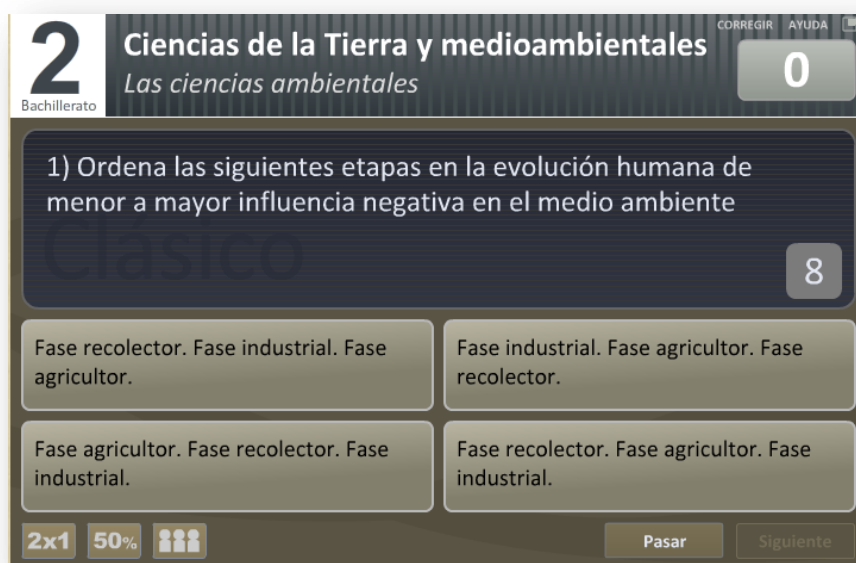


Figura 13: Captura de 'Testeando', interfaz de pregunta

Como se puede observar, la pantalla se encuentra dividida en 5 regiones claramente diferenciadas. En primer lugar, se observa el marcador de juego en la esquina superior derecha, que irá acumulando los puntos conseguidos hasta llegar al final de la partida. Más abajo, se disponen la parte que muestra el enunciado de la pregunta y el contador de tiempo transcurrido. Después, se encuentra la zona encargada de mostrar las 4 respuestas posibles, de las cuáles el usuario elegirá sólo una. A continuación, se muestran los comodines en la esquina inferior izquierda, cuya funcionalidad permitirá al usuario resolver las preguntas de una forma más sencilla. Finalmente, se encuentran los botones habilitados para pasar sin responder o para avanzar a la siguiente pregunta del test.

Testeando cuenta con multitud de modos de juego que, aunque se basen en la mecánica de juego explicada anteriormente, poseen diferentes normas y formas de puntuar. Entre las diferentes modalidades podemos encontrar el juego de tipo 'Tiempo muerto', que como su propio nombre indica omite la cuenta atrás de tiempo para responder la pregunta, o 'Probando', en el que la partida no termina hasta que se han respondido correctamente las diez preguntas planteadas.

2.3.7. Otras aplicaciones similares

A continuación se van a nombrar y describir brevemente otras herramientas y juegos similares. Estas aplicaciones no han sido incluidas en el análisis por su escasa aceptación, ya que, o son imitaciones de otras analizadas o poseen alguna característica que hacen que dicho análisis no sea del todo relevante.

En primer lugar se encuentra **Trivial Pursuit Online**¹⁰, forma versionada del juego de mesa original Trivial Pursuit que permite a los usuarios jugar partidas online y registrar sus puntuaciones. Al igual que el juego original, cada partida se basa en recorrer un tablero con forma de rueda respondiendo las preguntas según la temática de las casillas. Dicho tablero estará formado por 5 casillas especiales, que recompensarán al usuario con un 'quesito' si responde la pregunta de forma correcta. El juego tiene como principal objetivo llegar a conseguir 6 'quesitos'. Por otro lado, la aplicación muestra en todo momento las estadísticas de juego y las puntuaciones obtenidas por todos los usuarios dispuestas en rankings de diferentes tipos. La interfaz de juego es la siguiente:

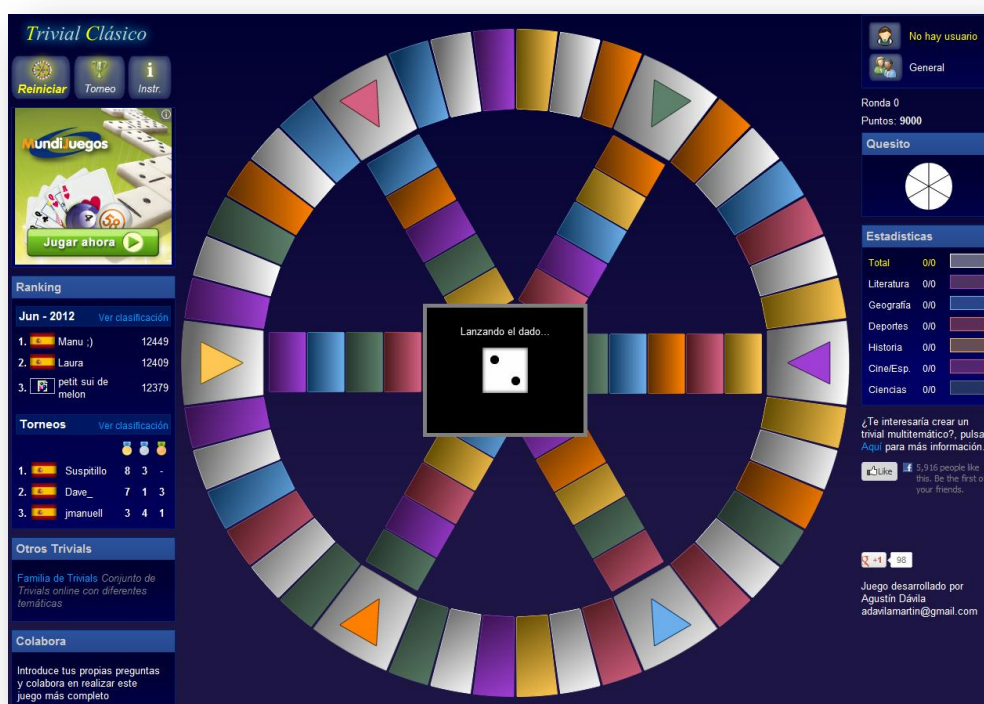


Figura 14: Captura de la interfaz de 'Trivial Pursuit Online'

¹⁰ Alojada en el sitio web <http://www.adavila.es>

Otra de las aplicaciones similares es **Quiz Millonario**¹¹, una versión del programa televisivo '¿Quién quiere ser millonario?' disponible para los usuarios de Tuenti. Cada una de las partidas se sucede hasta que el usuario falla alguna de las preguntas propuestas, en cuyo caso se brinda la posibilidad de compartir la puntuación obtenida en un ranking. Las preguntas no se encuentran clasificadas por temas y solo cuentan con una respuesta correcta entre cuatro posibles. Los usuarios contarán con tres comodines a lo largo de la partida, siendo estos un aumento de tiempo, paso de pregunta o el del 50% (encargado de descartar dos respuestas de las cuatro posibles). La interfaz de juego es la siguiente:

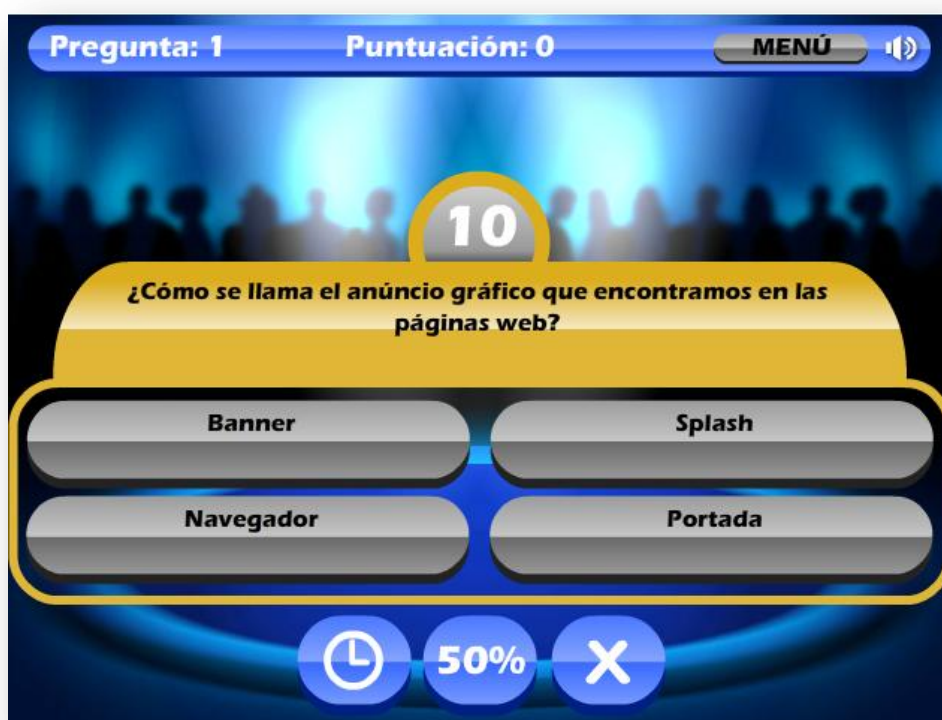


Figura 15: Captura de la interfaz de 'Quiz Millonario'

Y por último, **Triviador**¹², un juego que pueden disfrutar los usuarios de Facebook basado en la conquista de territorios de un mapa a otros usuarios a través de la respuesta de preguntas. Al igual que en varios juegos mencionados anteriormente, el sistema de preguntas se basa en acertar una de las cuatro respuestas posibles sin exceder el límite de tiempo. El transcurso de la partida se divide en dos partes diferenciadas: una donde los usuarios tienen que colocar a sus soldados por el mapa y otra donde tienen que competir por destruir el castillo contrario y conquistar el mayor número de territorios. En la primera fase, los jugadores que respondan de forma correcta a la pregunta ganarán uno o dos territorios dependiendo del tiempo empleado en responder. En la segunda fase, dos de los jugadores pelean por sus

¹¹ Alojada en <http://www.tuenti.com/>

¹² Alojado en <https://apps.facebook.com/triviador/>

Answer2Pass – Juego de evaluación interactiva de alumnos

territorios, quién acierta lleva a cabo la conquista y quien pierde es derrotado y se ve obligado a ceder su territorio o a dar por finalizada la partida por la pérdida del castillo. Cabe destacar la complejidad de la interfaz de juego y los efectos que se suceden durante la partida. Dicha interfaz es la siguiente:

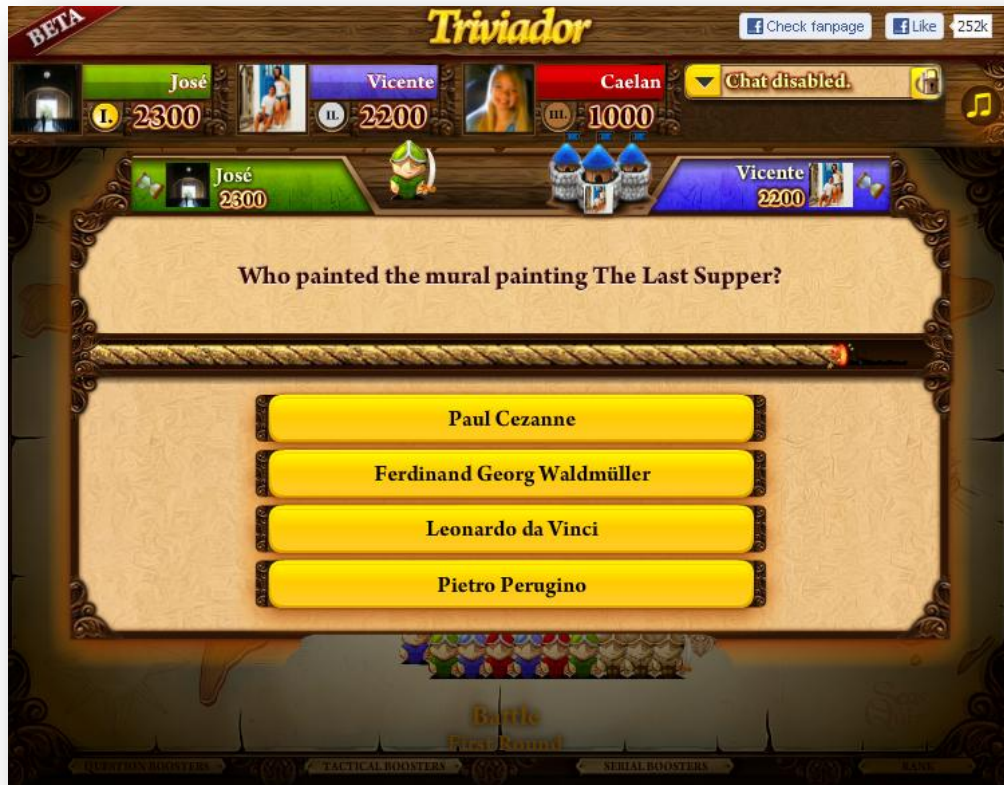


Figura 16: Captura de la interfaz de 'Triviador', Facebook

2.3.8. Comparación de resultados

A continuación se muestra una tabla con los objetivos que debería cumplir una aplicación que persiga el mismo fin que la desarrollada en el proyecto. El objetivo de esta matriz es comparar las aplicaciones similares a Answer2Pass para saber si pueden suplir o no su funcionalidad.

Aplicaciones Objetivos	Quiz Tool	Millonaire	Battery/Coffee Pot	Trivial Pursuit (Ipad)	Quizazz	Testeando
Juego educativo	✓	✓	✓	✓	✗	✓
Online	✓	✓	✓	✓	✓	✓
Gratuito	✓	✓	✓	✗	✓	✓
Interfaz llamativa	✗	✗	✓	✓	✗	✓
Interacción alumno-alumno	✗	✗	✗	✗	✗	✓
Editable	✓	✓	✓	✗	✓	✓
Clasificación por temas	✓	✗	✗	✓	✗	✓
Modos de pregunta	✓	✗	✗	✓	✗	✓
Baja complejidad	✓	✓	✓	✓	✓	✓
Ameno	✓	✓	✓	✓	✓	✓
Implementado en red social	✗	✗	✗	✗	✓	✗

Tabla 1: Comparación de resultados según los objetivos

Como se puede observar en la tabla anterior no se ha encontrado ninguna aplicación que actualmente pueda suplir todos los objetivos que se proponen conseguir con Answer2Pass. Cabe destacar que la herramienta web ‘Testeando’, es la que más se acerca a la finalidad perseguida, incumpliendo sólo el requisito de la implementación en alguna red social. Por otro lado, entrando en un análisis más exhaustivo, ‘Testeando’ tiene la deficiencia de estar traducida sólo al español, cosa que puede llegar a limitar el acceso a muchos usuarios. De la misma forma, dicha aplicación no comparte de ninguna manera las puntuaciones obtenidas por los usuarios, lo que impide cumplir el objetivo de agregar cierto grado de competitividad entre alumnos perseguido por Answer2Pass.

2.4. Análisis de tecnologías y alternativas

A continuación se van a describir todas las tecnologías candidatas para implementar la aplicación, explicando sus características, sus ventajas e inconvenientes, su instalación etc. De esta forma se recopilarán una serie de datos que servirán en el próximo apartado para poder identificar cuáles son las herramientas más adecuadas para la implementación de la aplicación.

2.4.1. Plataforma

En este apartado se va a proceder a realizar un análisis de las posibles plataformas en las que se puede implementar la aplicación a desarrollar. Las candidatas son las siguientes:

2.4.1.1. Facebook

Como bien es sabido, Facebook tiene la ventaja de dar a los usuarios la posibilidad de desarrollar aplicaciones que se muestren en la red social. De esta forma se puede crear una aplicación que use las librerías proporcionadas para potenciar la interacción de todos los usuarios de la red social compartiendo contenido unos con otros.

Una aplicación Facebook es un tipo de *iframe* cargado en el llamado 'Canvas Page', que será la parte de la red social donde se visualizará el contenido de la aplicación a desarrollar. En él se cargará todo el contenido Html, JavaScript y Css presente en la dirección web que alberga la aplicación. El 'Canvas Page' contará con una anchura de 760px por defecto, aunque se podrá configurar junto con la altura para que sea de tipo fluído.

Hay un amplio abanico de lenguajes de programación disponibles para la implementación, entre ellos se encuentran PHP, Java, Python o C++.

Los pasos para desarrollar una aplicación en Facebook son:

- 1- Pertenecer a la red social.
- 2- Agregar el desarrollador de aplicaciones



Figura 17: Captura de la pantalla de permisos de desarrollador, Facebook

Answer2Pass – Juego de evaluación interactiva de alumnos

- 3- Verificar la cuenta de usuario de tres formas posibles:
 - a. Vía sms
 - b. Agregando una Red y verificándola
 - c. Enviando el DNI con el nombre y fecha de nacimiento visibles
- 4- Configurar la aplicación especificando el nombre de la misma y la dirección web que debe ser cargada en el canvas.

Una vez hecho esto, Facebook proporcionará al desarrollador dos identificadores, el 'App ID' y el 'App Secret' que servirán para poder interactuar con las librerías y métodos de Facebook.

Facebook cuenta con 4 bibliotecas de programación oficiales que son Android, IOS, JavaScript y PHP, siendo esta última sobre la que más ejemplos e información se puede encontrar. Con estas librerías se puede acceder a métodos que a través de peticiones a Facebook (Realizadas gracias a los identificadores) son capaces de gestionar permisos, enviar notificaciones, publicar acciones en el tablón, recuperar información de los usuarios y muchas otras acciones interesantes para una aplicación.

Por otro lado, también existen librerías no oficiales para programar aplicaciones en Facebook, como por ejemplo RestFb¹³ y facebook-java-api¹⁴ para el lenguaje de programación Java.

2.4.1.2. Moodle

Como se mencionó en el apartado 1.3.1 Moodle es una plataforma usada principalmente por profesores para generar de forma sencilla material educativo que los alumnos puedan usar.

Este sistema de aprendizaje virtual tiene licencia Pública GNU y se distribuye de forma gratuita al ser una aplicación Open Source. Esto significa que cuenta con derechos de autor pero que el usuario puede utilizar con una serie de libertades, siempre y cuando acepte compartir su código con todos los usuarios, no modifique ni elimine el copyright y licencia originales y haga uso de esta para cualquier aplicación derivada.

Moodle es conocido en todo el mundo por ser una herramienta capaz de brindar a los profesores la oportunidad de promover y gestionar el aprendizaje de sus alumnos, de forma que pueden crear una aplicación web dinámica con las mejores herramientas a su disposición.

En cuanto a la interoperabilidad, Moodle puede funcionar en sistemas como Unix, Linux, FreeBSD, Windows, Mac OS X o NetWare, y cualquier otro que soporte

¹³ Para más información acceder a [23]

¹⁴ Para más información acceder a [8]

PHP y bases de datos. La instalación del mismo comprende el código Moodle ejecutado en un servidor PHP (Apache), una base de datos gestionada por MySQL, PostgreSQL, Microsoft SQL Server u Oracle; y un almacén de archivos para la subida de estos. Una vez instalado se copia automáticamente el código a un servidor web y se crea una base de datos en blanco. De la misma forma, Moodle también da la posibilidad de autoactualizar a siguientes versiones.

Para permitir su crecimiento, Moodle se estructura de forma modular, de manera que el núcleo se encuentra rodeado de una gran cantidad de plugins que los usuarios pueden consultar y editar. Físicamente los plugins son carpetas de scripts en PHP, junto con el CSS y Javascript necesarios, que se comunican con el núcleo a través de puntos de entrada definidos en algunas ocasiones en el archivo lib.php así como haciendo uso de diferentes librerías.

Algunos de los plugins más destacados que se pueden encontrar en Moodle son: Actividades, temas, paquetes de idiomas, formatos de cursos, esquemas de bases de datos, modelos de autenticación, sistemas de inscripción, repositorios etc.

En cuanto a las bases de datos, Moodle cuenta con gran cantidad de tablas, tanto por parte del núcleo como por parte de cada plugin desarrollado. La estructura de la misma se encuentra definida en el archivo install.xml dentro de la carpeta db de cada plugin. De esta forma en el archivo XML mod/course1/db/install.xml se encuentra la definición de la base de datos del curso 1 y en este otro lib/db/install.xml se encuentran definidas las tablas usadas por el núcleo de Moodle. Este archivo cuenta con comentarios que explican el propósito de cada tabla.

Una de las ventajas que ofrece Moodle es la facilidad de comunicación alumno-profesor y alumno-alumno, así como la posibilidad de actualización a nuevas versiones de forma muy sencilla. Por otro lado, posee gran cantidad de herramientas y recursos didácticos que permiten realizar aplicaciones dinámicas de forma gratuita y sencilla y, muy importante, es de tipo *Open Source*, que lo hace escalable y que favorece a los usuarios a la hora de implementar sus aplicaciones. La interfaz es sencilla, eficiente y compatible con todos los navegadores, disponiendo de variedad temas y plantillas que permiten personalizar el sitio web. Cabe destacar que también cuenta con programación orientada a objetos. En cuanto al seguimiento, registra el acceso de los alumnos, disponiendo de informes y gráficas de la actividad y participación de cada uno y permite la monitorización del desarrollo de todas las actividades. Por último, cuenta con traducciones a más de 70 idiomas y emplea mecanismos de seguridad en todas sus plataformas (validación de formularios, encriptación de cookies etc.).

Por otro lado, Moodle presenta desventajas como mostrar el mismo contenido a todos los alumnos, lo que significa que el proceso de aprendizaje no puede adecuarse al nivel y perfil de cada uno. También, se puede hacer complicado obtener soporte de la aplicación, ya que casi toda la información está en el foro escrita en inglés, lo que dificulta que usuarios sin conocimiento de este idioma obtengan solución a sus problemas.

2.4.2. Tecnologías de despliegue

A continuación se analizan las diferentes tecnologías sobre las que se puede realizar el despliegue de la aplicación. Las tecnologías candidatas son Google App-Engine y Amazon EC2.

2.4.2.1. Google App-Engine

Google AppEngine es una herramienta proporcionada por Google a sus usuarios que les permite implementar aplicaciones web alojándolas en su infraestructura. De esta forma crear una aplicación se convierte en algo sencillo, fácil de mantener y de escalar aumentando la capacidad de datos permitida. No hace falta mantener ningún tipo de servidor ni comprar ningún dominio, solo basta con subir la aplicación para que otros usuarios puedan disfrutarla.

La aplicación se puede configurar de tal modo que todo el mundo pueda acceder a ella o que tenga acceso limitado por el usuario.

Si ya se posee un dominio, también se puede almacenar la aplicación, en caso contrario se le puede asignar uno gratuito terminado en *appspot.com*.

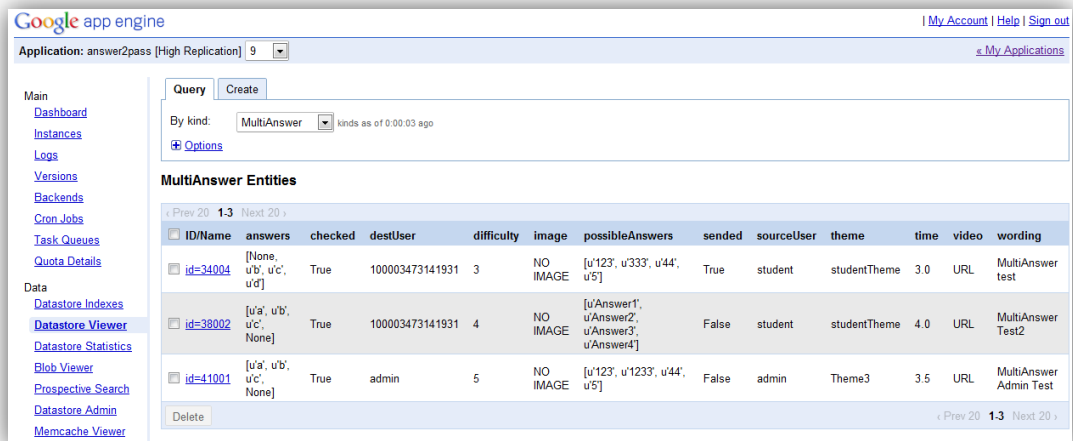
Google App Engine soporta varios lenguajes de programación. Entre ellos se encuentra Java, haciendo uso de *Java runtime enviroment* se pueden utilizar todas las herramientas que este lenguaje ofrece, como JVM, Servlets etc; Python, contando con dos *Python runtime enviroments* dedicados; y por último, también proporciona un entorno para el lenguaje de programación Go.

Por ejemplo, si queremos desarrollar una aplicación Web con App Engine usando Java en un entorno de desarrollo eclipse tendríamos que hacer lo siguiente:

- 1- Poseer una cuenta de Google.
- 2- Instalar Java y el plugin de Google App Engine para Eclipse.
- 3- Crear un proyecto y desarrollar la aplicación.
- 4- En las propiedades del proyecto establecer la versión y hacer click en Deploy para subir la aplicación al servidor.
- 5- Probar la aplicación con el dominio especificado y gestionar con la consola de administrador.

Como se ha mencionado anteriormente, gestionar una aplicación desarrollada con Google App Engine es muy sencillo. Google proporciona al usuario una consola donde puede controlar la versión visible a los usuarios, la actividad y las peticiones producidas sobre la aplicación, acceder y modificar a los datos almacenados, administrar los permisos y la configuración de la cuenta etc.

Answer2Pass – Juego de evaluación interactiva de alumnos



The screenshot shows the Google App Engine Admin Console for the application 'answer2pass'. The 'MultiAnswer Entities' table is displayed, showing a list of entities with columns: ID/Name, answers, checked, destUser, difficulty, image, possibleAnswers, send, sourceUser, theme, time, video, and wording. The table contains three rows of data.

ID/Name	answers	checked	destUser	difficulty	image	possibleAnswers	send	sourceUser	theme	time	video	wording
[id=34004]	[None, u'b', u'c', u'd']	True	100003473141931	3	NO IMAGE	[u'123', u'333', u'44', u'5']	True	student	studentTheme	3.0	URL	MultiAnswer test
[id=38002]	[u'a', u'b', u'c', None]	True	100003473141931	4	NO IMAGE	[u'Answer1', u'Answer2', u'Answer3', u'Answer4']	False	student	studentTheme	4.0	URL	MultiAnswer Test2
[id=41001]	[u'a', u'b', u'c', None]	True	admin	5	NO IMAGE	[u'123', u'1233', u'44', u'5']	False	admin	Theme3	3.5	URL	MultiAnswer Admin Test

Figura 18: Interfaz de administrador, Google App Engine

Cabe destacar que Google App-Engine hace uso de JDO como forma de almacenamiento de datos.

2.4.2.2. Amazon EC2

Amazon EC2 (Elastic Compute Cloud), es un servicio que ofrece Amazon Web Services que ofrece almacenamiento con tamaño variable en la nube. Principalmente está diseñado para facilitar recursos para la implementación de aplicaciones Web.

Amazon proporciona una sencilla interfaz que permitirá al usuario configurar la capacidad del servicio o incrementarla, así como la posibilidad de administrar los recursos utilizados.

Para utilizar Amazon EC2 el usuario necesita:

- Seleccionar una plantilla de las preconfiguradas o crear una AMI (Amazon Machine Engine).
- Realizar la configuración relativa a la seguridad y al acceso de la instancia.
- Seleccionar los tipos de instancia, entre los que podemos encontrar instancias estándar, microinstancias, instancias con gran cantidad de memoria, instancias para CPU de alto rendimiento, instancias informáticas en clúster o instancias de GPU para clusters;
- Seleccionar el sistema operativo deseado e iniciar, finalizar y supervisar todas las instancias del AMI del usuario necesarias.
- Optar por la opción gratuita o pagar por los recursos necesarios.

Amazon EC2 soporta gran variedad de lenguajes de programación de los que ofrece librerías, entre ellos se encuentra Android, IOS, Java, .NET, PHP y Ruby. De la misma forma, se ofrece a los usuarios herramientas para desarrollar en entornos de programación como Visual Studio o Eclipse.

2.4.3. Tecnologías de almacenamiento

Finalmente se procede a realizar el análisis de las tecnologías candidatas para llevar a cabo el almacenamiento de datos de la aplicación a desarrollar. Estas son las siguientes:

2.4.3.1. JDO

JDO proviene de Java Data Object, y es una interfaz estándar para almacenar objetos Java en una base de datos. Permite que las interfaces realicen tareas de anotación de objetos Java, de recuperación de objetos con consultas y de interacción con una base de datos mediante transacciones. A través de una serie de funciones y las consultas mencionadas se pueden recuperar, modificar y eliminar datos como si de una base de datos MySQL tradicional se tratara. La diferencia se encuentra en que JDO puede trabajar con distinto tipo de bases de datos sin usar ningún lenguaje específico.

2.4.3.2. MySQL

MySQL es un sistema gestor de base de datos relacional, multihilo y multiusuario. Esto significa que los modelos de datos están basados en la lógica de predicados y en la teoría de conjuntos, que permite realizar varias tareas a la vez y que tiene la capacidad de proveer servicio a múltiples usuarios simultáneamente.

Las bases de datos MySQL son accesibles desde aplicaciones escritas en diversos lenguajes de programación. Entre ellos se encuentran C, C++, C#, Java, Pascal, Perl, PHP, Python etc.

MySQL soporta gran cantidad de datos. Por ejemplo, MySQL Server cuenta con bases de datos de hasta 50 millones de registros. Dicha bases de datos estarán formadas por tablas relacionadas entre sí, y estas a su vez por campos en los que los datos serán almacenados. Estos datos podrán ser recuperados, borrados y modificados por medio de consultas.

En cuanto a la seguridad, MySQL cuenta con un sistema seguro de contraseñas y privilegios que regulan el acceso de los usuarios a los datos, evitando así que estos caigan en malas manos y puedan ser usados para fines no autorizados.

2.5. Selección de tecnologías

Una vez que se han analizado todas las tecnologías candidatas, se pasará a dar en este apartado una serie de razones determinantes para la elección final tomada en cuanto a las tecnologías a usar en la aplicación.

Para comenzar, en cuanto a la plataforma se refiere, se ha decidido implementar la aplicación en Facebook en vez de en Moodle. Una de las razones que han determinado dicha elección ha sido que al ser Facebook una red social se potencia el uso de la aplicación por un mayor número de usuarios. Por ello, estos tendrán mayor tendencia a jugar una partida de Answer2pass mientras consultan su red social antes que a través de un enlace externo. Por otro lado, las notificaciones y publicación de resultados en el tablón son un punto a favor, ya que aportan algo más de competitividad al juego y hacen que este se comparta. También hay que tener en cuenta que la estructura de Moodle tiene un grado de complejidad que requiere más tiempo de estudio que el que se posee para la implementación de la aplicación. Y por último, Moodle limita el lenguaje de programación a usar, ya que sólo está disponible para PHP.

Por otro lado, se ha decidido implementar la aplicación haciendo uso del servicio que ofrece Google App Engine. Una de las razones que justifican esta decisión ha sido que, de esta forma, no es necesario un coste adicional para la compra de servidor y dominio. Por otro lado, Amazon EC2 ofrece una cantidad de tiempo gratuito limitado, es decir, al sobrepasar un determinado número de horas se debe aplicar un coste adicional (Google App Engine no tiene límite de tiempo gratuito). Otra de las razones que justifican dicha decisión es que Google ofrece una gestión de la aplicación muy sencilla, siendo de baja complejidad la instalación del entorno de trabajo. Debido a dicha simplicidad de instalación, no es necesario descargar e instalar gestores de bases de datos ni servidores independientes, tarea que suele dar diversos conflictos y puede suponer una pérdida de tiempo en lo que a la implementación de la aplicación se refiere. Por otro lado, queda descartada la opción de MySQL como sistema de almacenamiento de datos (Google App-Engine no hace uso de ello), por lo que se utilizará JDO. En cuanto a dicha tecnología, aunque se tenga poca experiencia con ella, proporciona una mayor facilidad a la hora de consultar y almacenar datos. El lenguaje de programación, entre los que ofrece Google App Engine, será Java. Esta decisión ha sido tomada en base a la inexperiencia que se posee con los restantes. Al usar Java como lenguaje de programación se obtiene beneficio por el uso de *servlets*. Al emplearlos se produce un ahorro de recursos del sistema, ya que se ejecutan de forma paralela usando *threads* y no procesos. Es una tecnología muy fácil de implementar que dará dinamismo a la aplicación.

Una vez tomadas todas estas decisiones, se debe tomar una última ¿Qué API de Facebook se usará en la aplicación? La respuesta se limita, por el lenguaje de programación utilizado, a Java (Versión no oficial) o JavaScript, descartando de esta forma a IOS y PHP.

Finalmente se ha decidido usar el SDK de JavaScript para realizar la comunicación con Facebook en la aplicación. Una de las razones que han permitido tomar esta decisión ha sido la experiencia con el lenguaje, así como la baja complejidad y la cantidad de información de ayuda que Facebook facilita sobre la misma (Incluidos ejemplos de gran utilidad). Se podría haber empleado la biblioteca de Java, pero debido a la no oficialidad de la misma se produjeron diversos conflictos a la hora de realizar la implementación.

2.6. Diagrama de casos de uso

A continuación se van a incluir los diagramas de caso de uso para los diferentes actores que pueden hacer uso de la aplicación, en este caso administrador (profesor) y usuario (alumno). De esta forma se representará el comportamiento del sistema ante la actuación de las entidades administrador (Profesor) y usuario (Alumno). Finalmente se dará una breve explicación de cada uno de ellos.

Los casos de uso estarán relacionados mediante asociaciones. Cuando esta unión se realice mediante una flecha discontinua acompañada del término '<<include>>' significará inclusión, o lo que es lo mismo, que la funcionalidad de un caso de uso requiere la funcionalidad de otro anterior. Por otro lado, cuando la unión se realice mediante una flecha discontinua acompañada del término '<<extend>>' significará extensión, es decir, que un caso de uso agrega funcionalidad a otro. Por último, cuando dos casos de uso se encuentren unidos por una flecha formada por una línea continua significará generalización, que supondrá una relación padre-hijo.

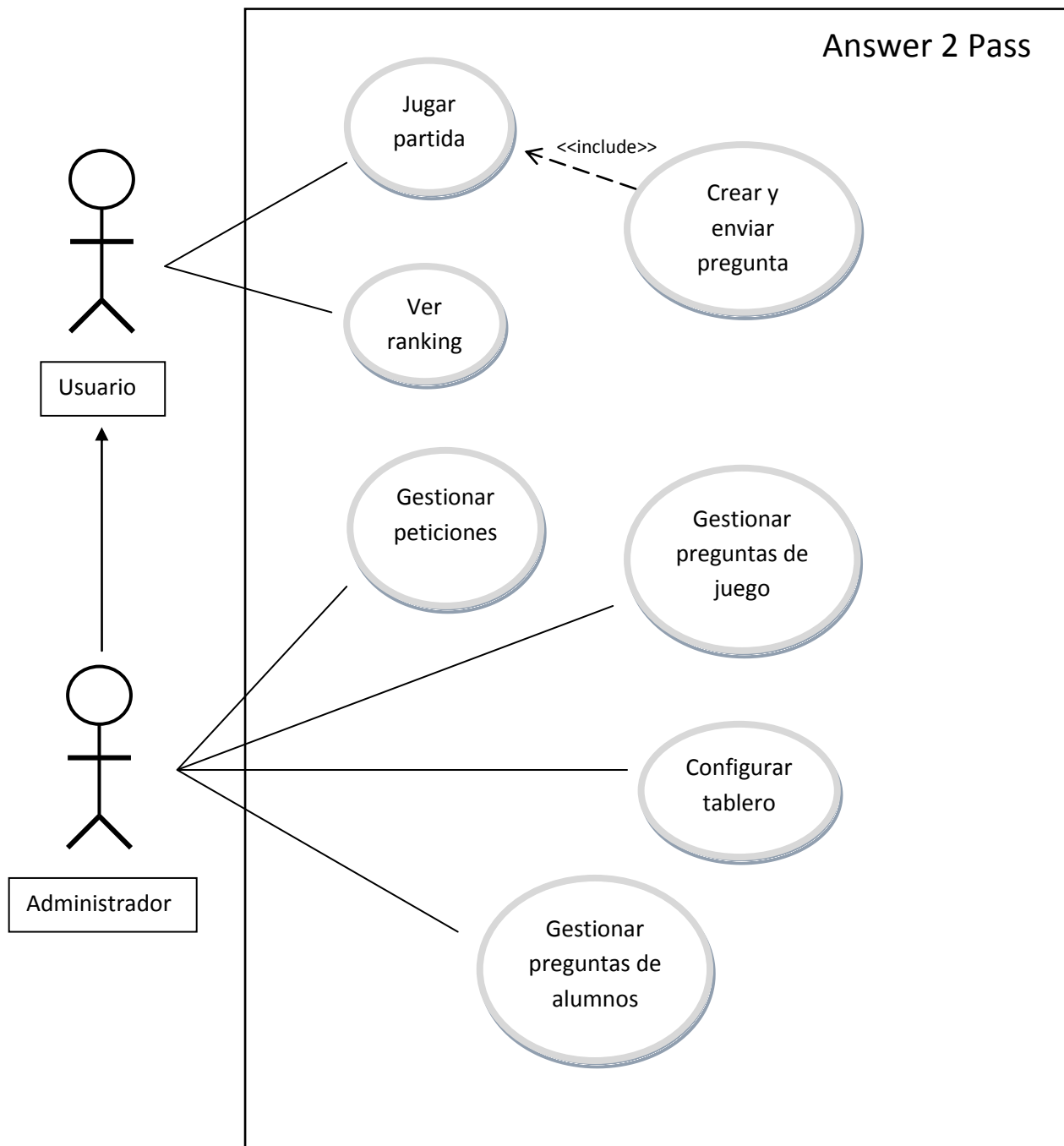


Figura 19: Diagrama de casos de uso

Como se puede observar en el diagrama anterior, los casos de uso identificados en el sistema han sido los siguientes:

Jugar Partida: Como su propio nombre indica, a través de este caso de uso se representa la acción de jugar partida por parte del usuario o del administrador. En el desarrollo de la partida, los jugadores irán respondiendo las preguntas propuestas por los profesores a lo largo de todo el tablero hasta llegar al final del mismo. Para conseguir mejor puntuación podrán ayudarse de fichas bonus, que sólo podrán adquirir con la respuesta correcta a las preguntas recibidas por otros alumnos.

Crear y enviar pregunta: Se trata de un caso de uso que se encuentra incluido dentro de 'Jugar Partida', por lo que sólo se puede llevar a cabo si una partida está en curso. Es la acción que lleva a cabo un jugador cuando su ficha de juego cae en la casilla comodín habilitada con esta funcionalidad. Par avanzar en el tablero, el jugador se verá obligado a crear una nueva pregunta, que será enviada a otro compañero después de ser comprobada por el profesor.

Ver Ranking: Posibilidad de visualizar las puntuaciones obtenidas por los jugadores dispuestas de forma descendente en un ranking de juego. Los tipos de ranking que se pueden consultar son: Mejor respondedor, mejor preguntador, puntuación media, mejor puntuación y puntuación de la última partida.

Gestionar peticiones: Acción llevada a cabo por el administrador que permite aceptar rechazar y notificar las peticiones de acceso al juego enviadas por los usuarios.

Gestionar preguntas de juego: Acción propia del administrador que permite crear nuevas preguntas de juego y eliminar las ya creadas almacenadas en el banco de preguntas.

Gestionar preguntas de alumnos: Acción llevada a cabo por el administrador que permite corregir las preguntas enviadas entre alumnos. De esta forma, el profesor podrá eliminar y verificar preguntas enviadas

Configurar tablero: Acción por la que el administrador puede generar nuevos tableros o gestionar los ya creados. La gestión de tableros incluye la visualización y borrado de los mismos, así como la posibilidad de elección de tablero de juego visible a los alumnos.

2.7. Requisitos de software

A continuación se van a especificar los requisitos de software, que valdrán para dar una descripción completa de la aplicación a desarrollar y del comportamiento de la misma. Para comenzar se especificará que formato siguen las tablas de requisitos y posteriormente se pasará a establecer cuáles son los requisitos funcionales y no funcionales de la aplicación. Los primeros nos valdrán para saber cómo se comportará el software y serán complementados con los segundos, los requisitos no funcionales, que determinarán las restricciones en el diseño o implementación de la aplicación.

2.7.1. Formato seguido en la especificación de requisitos

Las tablas de requisitos estarán compuestas por:

- **Identificador:** Clave única que identifica cada requisito. Estará compuesta por las siglas de Requisito Funcional (RF) o Requisito No Funcional (RNF) seguido de un número único asignado de dos cifras.
- **Nombre:** Nombre del requisito. Debe ser significativo en relación a la descripción.
- **Descripción:** Explicación detallada del requisito.
- **Prioridad:** Nivel de preferencia que tiene el requisito para el equipo de desarrollo.
- **Verificabilidad:** Apartado formado por los identificadores de las pruebas que verifican el cumplimiento del requisito.

2.7.2. Requisitos funcionales

A continuación se especifican los requisitos funcionales, necesarios para entender la aplicación y su comportamiento. Estos se dispondrán en dos grupos diferenciados, los requisitos de operación y los de datos.

El modelo que deben seguir las tablas de los requisitos funcionales es la siguiente:

Identificador	Nombre	Descripción	Prioridad	Verificabilidad
RF – XX			Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja <input type="checkbox"/>	

Tabla 2: Formato de la tabla de requisitos funcionales

2.7.2.1. Requisitos funcionales de operación

Identificador	Nombre	Descripción	Prioridad	Verificabilidad
RF - 01	Petición de registro	La aplicación debe dar la posibilidad al usuario de enviar una petición al profesor ¹⁵ para poder acceder al juego.	Alta	PA - 02
RF - 02	Permisos	La aplicación debe proporcionar al usuario la opción de aceptar los permisos necesarios para poder jugar.	Alta	PA - 05
RF - 03	Límite de acceso	La aplicación debe limitar el acceso sólo a aquellos jugadores cuyas peticiones han sido aceptadas y que han proporcionado los permisos necesarios.	Alta	PA – 03, PA – 04, PA - 05
RF - 04	Tipos de ranking	Los rankings de la aplicación deben ser ‘Mejor preguntador’, ‘Mejor respondedor’, ‘Puntuación última partida’, ‘Puntuación media’ y ‘Mejor puntuación’.	Alta	PA - 06
RF - 05	Fichas bonus	La aplicación debe permitir al usuario hacer uso de fichas bonus. A través de ellas el usuario puede realizar una tirada de 2 posiciones.	Alta	PA - 20
RF - 07	Tiempo límite	Cada pregunta planteada debe tener un tiempo límite.	Alta	PA - 10
RF - 08	Corrección de respuesta	La aplicación debe mostrar un mensaje indicando si la pregunta se ha respondido correctamente o no.	Alta	PA - 11
RF - 09	Preguntas no respondidas de otros usuarios	La aplicación debe permitir al usuario visualizar las preguntas enviadas por otros alumnos mostrando el tipo de pregunta (True/False, una sola respuesta y multirespuesta) y la dificultad de la misma.	Alta	PA – 18, PA - 19
RF - 10	Responder pregunta de usuario	Una vez se ha respondido de forma correcta una pregunta recibida de otro usuario, el jugador conseguirá una de las fichas mencionadas en el requisito RF - 05. De la misma forma, se recalculará su puntuación de ‘Mejor respondedor’ según la dificultad de la misma.	Alta	PA – 18, PA - 19

¹⁵ El profesor posee el rol de administrador, por lo tanto, aunque se mencionen ambos términos a lo largo de los requisitos y las pruebas, poseen el mismo significado para la aplicación.

Answer2Pass – Juego de evaluación interactiva de alumnos

Identificador	Nombre	Descripción	Prioridad	Verificabilidad
RF - 11	Comodines de juego	La aplicación será capaz de bonificar al usuario cada vez que caiga en una casilla comodín. Los comodines disponibles serán de doble tirada, doble tiempo, comodín aleatorio, creación de pregunta y tema aleatorio.	Alta	PA – 12, PA – 13, PA – 14, PA – 15, PA – 16
RF - 12	Publicación de puntuación	La aplicación debe permitir al usuario publicar la puntuación conseguida en su muro de Facebook una vez finalizada la partida.	Media	PA - 22
RF - 13	Cálculo de puntuaciones	Una vez finalizada la partida, la aplicación debe emplear la puntuación obtenida para calcular los resultados de los rankings ‘Mejor puntuación’, ‘Puntuación media’ y ‘Puntuación última partida’ mencionados en el requisito RF-04.	Alta	PA - 21
RF - 14	Reinicios necesarios cada nueva partida	Cada vez que un usuario acabe o comience una partida se establecerá la ficha en la primera casilla del tablero y se reiniciarán la puntuación, el tiempo y las fichas de bonus.	Alta	PA - 23
RF - 15	Amplitud de la tirada	La aplicación debe permitir al usuario ampliar el número de avances por el tablero dependiendo del tiempo que haya empleado en contestar la pregunta. Si el jugador acierta, la tirada irá de 2 a 6 según el intervalo de tiempo en el que se encuentre. Por el contrario, si falla solo avanzará una casilla.	Alta	PA - 11
RF - 16	Cálculo de posición en el tablero	La aplicación debe calcular la todas las posibles posiciones de la ficha cuando el usuario responda una pregunta (Según la amplitud de tirada definida en el requisito RF - 15), mostrando fichas con alguna diferencia representativa en las casillas posibles.	Alta	PA - 11
RF - 17	Cálculo de puntuación por pregunta	La aplicación debe poder sumar, al marcador descrito en el requisito RF - 34, el tiempo de respuesta y una tirada más de forma correcta cuando el usuario responda una pregunta.	Alta	PA - 11
RF - 18	Respuestas marcadas	El usuario debe poder saber que respuestas ha marcado como correctas, dando la posibilidad de cambiarlas si el tiempo no ha finalizado.	Baja	PA - 10
RF - 19	Creación de nueva pregunta	La aplicación presentará al usuario un editor de preguntas cada vez que caiga en la casilla comodín “Creación de pregunta”. Dicha pregunta será enviada a otro alumno asignado de forma aleatoria.	Alta	PA - 16

Answer2Pass – Juego de evaluación interactiva de alumnos

Identificador	Nombre	Descripción	Prioridad	Verificabilidad
RF - 21	Creación de pregunta por parte del administrador	La aplicación debe permitir al administrador crear nuevas preguntas de juego.	Alta	PA - 24
RF - 22	Banco de preguntas de juego	El administrador podrá visualizar el banco de preguntas de juego creadas por cualquiera de los administradores de la aplicación.	Alta	PA - 25
RF - 23	Banco de preguntas enviadas por alumnos	El administrador podrá visualizar, dentro de la pantalla de preguntas creadas por los alumnos, la lista de preguntas sin corregir y un historial de todas las preguntas ya enviadas y verificadas.	Alta	PA - 26
RF - 24	Enviar notificación	El administrador podrá enviar una notificación al Facebook del usuario que recibirá la pregunta creada por otro alumno.	Alta	PA - 29
RF - 25	Corregir y enviar	El administrador tendrá la posibilidad de corregir y enviar la pregunta creada por un alumno, aumentando así su puntuación de 'Mejor preguntador' según la calidad y la dificultad de dicha pregunta.	Alta	PA - 28
RF - 26	Eliminar preguntas	El administrador podrá eliminar todo tipo de preguntas, tanto las de usuarios como las creadas por los administradores.	Alta	PA - 30
RF - 27	Aceptar o rechazar peticiones	El administrador podrá aceptar o rechazar las peticiones de acceso al juego enviadas por los usuarios, así como enviar mensajes en forma de notificación al Facebook de los mismos.	Alta	PA – 32, PA - 33, PA - 34
RF - 28	Generar tableros	La aplicación debe ser capaz de aleatorizar los colores y comodines dispuestos en las casillas del tablero.	Alta	PA - 35
RF - 29	Tablero generado	La aplicación debe mostrar en una nueva pantalla el resultado del tablero generado.	Media	PA - 35
RF - 30	Visualización de los tableros generados	Al administrador podrá visualizar los tableros generados en una lista.	Alta	PA - 37
RF - 31	Borrado de los tableros generados	Al administrador podrá eliminar cualquier tablero generado previamente.	Alta	PA - 38

Answer2Pass – Juego de evaluación interactiva de alumnos

Identificador	Nombre	Descripción	Prioridad	Verificabilidad
RF - 32	Selección de tablero	Al administrador podrá establecer un nuevo tablero de juego seleccionando el elegido en la lista de tableros (RF-30).	Alta	PA - 39
RF - 50	Filtrado de preguntas	Todas las preguntas de los contenedores deben poder ser filtradas por tipo, dificultad y tema	Baja	PA - 40

Requisitos 1: Requisitos funcionales de operación

2.7.2.2. Requisitos funcionales de datos

Identificador	Nombre	Descripción	Prioridad	Verificabilidad
RF - 33	Menú inicial	La aplicación al iniciarse debe mostrar las instrucciones de acceso a la misma y el logo de la universidad y del programa.	Alta	PA - 01
RF - 34	Formulario de registro	Los campos del formulario de petición de registro deben ser nombre completo, NIA y grupo de clase.	Alta	PA - 01
RF - 35	Marcador	La aplicación debe mostrar un marcador que indique el número de tiradas y el tiempo empleado en responder las preguntas de la partida.	Alta	PA - 11
RF - 36	Menú principal del jugador	Una vez dentro de la aplicación, el menú principal de los jugadores debe mostrar las opciones de comenzar nueva partida, visualizar rankings o ver contenedor de preguntas de otros alumnos no respondidas.	Alta	PA - 07
RF - 37	Menú principal del administrador	Una vez dentro de la aplicación, el menú principal del administrador debe mostrar las opciones de gestionar preguntas de administrador, gestionar preguntas de alumnos, visualizar rankings, gestionar peticiones, gestionar tableros de juego y acceder al juego.	Alta	PA - 08

Answer2Pass – Juego de evaluación interactiva de alumnos

Identificador	Nombre	Descripción	Prioridad	Verificabilidad
RF - 38	Contenido de los rankings	Los rankings de tipo ‘Puntuación última partida’, ‘Puntuación media’ y ‘Mejor puntuación’ deben contener el nombre del usuario, el número de tiradas y el tiempo conseguido dispuestos en orden descendente. Los de tipo ‘Mejor preguntador’, ‘Mejor respondedor’ deben contener el nombre del usuario y los puntos acumulados dispuestos en orden ascendente. Además, el ranking ‘Puntuación media’ deberá contener otro campo que indique el número de partidas finalizadas.	Alta	PA - 06
RF - 39	Formato de pregunta	La pantalla de preguntas de la aplicación debe mostrar el tiempo restante, la barra de tiempo, la dificultad, el enunciado, vídeo (opcional), las respuestas posibles y la opción de corregir la respuesta.	Alta	PA - 10
RF - 41	Pantalla de fin de partida	La aplicación debe mostrar al usuario una pantalla al finalizar la partida que contenga el número de tiradas y la suma del tiempo utilizado en la respuesta de preguntas.	Alta	PA - 21
RF - 42	Tipos de pregunta	Las preguntas de juego tratarán de distinto tema (Establecido por el administrador) y serán de tipo True/False y con una sola o múltiples respuestas.	Alta	PA - 17
RF - 43	Campos de nueva pregunta de alumno	Los campos mostrados por la aplicación al crear nueva pregunta deben ser tipo, enunciado, dificultad, respuestas y tiempo.	Alta	PA - 16
RF - 44	Campos del creador de preguntas por parte del administrador	Para generar nuevas preguntas de juego, el administrador deberá completar los campos dispuestos en un generador de preguntas. Estos serán los mismos que los especificados en el requisito RF - 42 añadiendo vídeo (opcional) y el tema de la pregunta.	Alta	PA - 24
RF - 45	Contenido de las preguntas en los contenedores	Las preguntas pertenecientes a los contenedores definidos en los requisitos RF - 21 y RF - 22 deben mostrar el identificador de la pregunta, el usuario de origen, el usuario de destino, el tema, el tiempo, la dificultad, la URL del vídeo (opcional), el enunciado y las respuestas.	Alta	PA - 27
RF - 46	Lista de peticiones	El administrador podrá visualizar una lista con todas las peticiones enviadas por los alumnos. Dicha lista contendrá los campos nombre completo, NIA y grupo.	Alta	PA - 31

Answer2Pass – Juego de evaluación interactiva de alumnos

Identificador	Nombre	Descripción	Prioridad	Verificabilidad
RF - 47	Alumnos aceptados	El administrador podrá visualizar una lista con todas las peticiones aceptadas previamente. Dicha lista contendrá los campos nombre completo, NIA y grupo.	Alta	PA - 32
RF - 48	Colores y temas en tableros generados	El administrador podrá generar los tableros estableciendo el nombre del mismo y el nombre de los temas y colores que desee incluir.	Alta	PA - 35
RF - 49	Campos de la lista de tableros generados	La aplicación debe mostrar una lista de tableros cuyo campos sean el nombre del tablero, los temas, e iconos de visualización y borrado.	Alta	PA - 36
RF - 20	Dificultad	Las preguntas deben tener un grado de dificultad comprendido entre 1 y 5.	Media	PA - 10

Requisitos 2: Requisitos funcionales de datos

2.7.3. Requisitos no funcionales

A continuación se especifican los requisitos no funcionales de la aplicación, cuya finalidad es indicar restricciones del sistema. El modelo que deben seguir las tablas de dichos requisitos es el mismo empleado para especificar los requisitos funcionales del apartado anterior.

2.7.3.1. Requisitos no funcionales de rendimiento

Identificador	Nombre	Descripción	Prioridad	Verificabilidad
RNF – 01	Navegadores	La aplicación debe poder visualizarse y ejecutarse correctamente en los navegadores Google Chrome, Mozilla Firefox y Safari.	Alta	-

Requisitos 3: Requisitos no funcionales de rendimiento

2.7.3.2. Requisitos no funcionales de interoperabilidad

Identificador	Nombre	Descripción	Prioridad	Verificabilidad
RNF – 02	SDK de JavaScript	La aplicación debe usar el SDK de JavaScript proporcionado por Facebook para realizar la comunicación con el mismo.	Alta	-
RNF – 03	Anchura	La aplicación debe estar limitada a 760px de ancho, tamaño proporcionado por Facebook.	Alta	

Requisitos 4: Requisitos no funcionales de interoperabilidad

2.7.3.3. Requisitos no funcionales de interfaz

Identificador	Nombre	Descripción	Prioridad	Verificabilidad
RNF – 04	Tablero de juego	El tablero de juego debe estar compuesto por casillas de colores (Dependiendo del tema de la pregunta) y de tipo comodín (Definidos en el requisito RF - 11)	Alta	PA-35

Answer2Pass – Juego de evaluación interactiva de alumnos

RNF – 06	Formato de contadores de tiempo	El tiempo límite para responder las preguntas vendrá representado tanto de forma numérica como gráfica. Esto último se realizará mediante una barra de tiempo.	Media	PA-10
RNF – 07	Contenedor de fichas bonus	La aplicación debe mostrar un contenedor que almacene las fichas definidas en el requisito RF – 05.	Baja	PA-10
RNF – 08	Formato de la dificultad	La dificultad de las preguntas en la aplicación se medirá a través de estrellas.	Baja	PA-18,PA-19

Requisitos 5: Requisitos no funcionales de interfaz

2.7.3.4. Requisitos no funcionales de accesibilidad

Identificador	Nombre	Descripción	Prioridad	Verificabilidad
RNF – 05	Idioma	La aplicación debe mostrar todo su contenido en inglés	Alta	-

Requisitos 6: Requisitos no funcionales de accesibilidad

2.8. Plan de pruebas

A continuación se va a especificar un plan de pruebas de aceptación válidas para verificar todos los requisitos establecidos en el punto 2.7. Todas ellas deben ser superadas para considerar que el proyecto ha finalizado con éxito cumpliendo todos los objetivos establecidos.

Finalmente se mostrará una matriz que representará la relación entre los requisitos y las pruebas que verifican su funcionalidad.

2.8.1. Formato de las pruebas

Las tablas de las pruebas contendrán los siguientes campos:

- **Identificador:** Clave única que identifica cada prueba. Estará compuesta por las siglas de Prueba de Aceptación (PA) seguido de un número único asignado de dos cifras.
- **Nombre:** Nombre que identifica la prueba según su descripción.
- **Entrada esperada:** Dato o acción de entrada que debe proporcionarse para poder llevar a cabo la prueba.
- **Salida esperada:** Resultado de salida que debe obtenerse una vez realizada la prueba.

La tabla de pruebas tendrá la siguiente apariencia:

Identificador	Nombre	Entrada Esperada	Salida Esperada
PA - XX			

Tabla 3: Formato de la tabla de pruebas

2.8.2. Catálogo de pruebas

Identificador	Nombre	Entrada Esperada	Salida Esperada
PA - 01	Elementos correspondientes en la pantalla inicial de la aplicación	El usuario inicia Answer2Pass, dispuesta en la lista de aplicaciones de Facebook.	Se visualizan el logo de la aplicación, el de la universidad y las instrucciones para comenzar. De la misma forma se debe visualizar el formulario de envío de petición, que tendrá los campos nombre, NIA y grupo.
PA - 02	Envío correcto de petición	Envío de petición desde la pantalla inicial enviando los campos grupo, NIA y nombre.	El sistema muestra el siguiente mensaje de confirmación: <i>"Request sended correctly!"</i> .
PA - 03	Control de acceso I: Intento de acceso sin una petición confirmada o enviada	Click en el logo de la aplicación para acceder al juego. Realizar esto sin haber enviado una petición previamente, o sin que esta esté confirmada.	Si la petición no ha sido enviada, el sistema muestra el siguiente mensaje: <i>"Please, send a request to enter the game"</i> . Si ha sido enviada pero no confirmada, el sistema muestra: <i>"Your request is being processed"</i> .
PA - 04	Control de acceso II: Intento de acceso sin dar permisos	Click en el logo de la aplicación para acceder al juego. Realizar esto sin proporcionar los permisos necesarios a la aplicación.	El sistema muestra el siguiente mensaje: <i>"Please check permissions"</i> .
PA - 05	Control de acceso III: Intento de acceso con permisos y con petición confirmada	Click en el logo de la aplicación para acceder al juego. Realizar esto habiendo proporcionado los permisos necesarios y siendo la petición aceptada.	En caso de un usuario con rol de 'Alumno', se debe visualizar la pantalla de juego junto con el tablero. En caso de un usuario con rol de 'Profesor', se debe visualizar la pantalla de administrador que contiene la pila de preguntas de juego creadas.

Answer2Pass – Juego de evaluación interactiva de alumnos

Identificador	Nombre	Entrada Esperada	Salida Esperada
PA - 06	Tipos de ranking y campos correctos	Hacer click en la opción 'Rankings' del menú.	Comprobar que los rankings visualizados son: 'Mejor preguntador', 'Mejor respondedor', 'Puntuación última partida', 'Puntuación media' y 'Mejor puntuación'. Comprobar a su vez que los de tipo 'Puntuación última partida', 'Puntuación media' y 'Mejor puntuación' contienen el nombre del usuario, el número de tiradas y el tiempo conseguido dispuestos en orden descendente; los de tipo 'Mejor preguntador', 'Mejor respondedor' contienen el nombre del usuario y los puntos acumulados dispuestos en orden ascendente; y por último, que el ranking 'Puntuación media' contiene otro campo que indica el número de partidas finalizadas.
PA - 07	Contenido del menú principal de jugador	Haber accedido al juego previamente con rol 'Alumno'.	El menú principal del jugador debe mostrar las siguientes opciones: comenzar nueva partida, visualizar rankings o ver contenedor de preguntas de otros alumnos no respondidas.
PA - 08	Contenido del menú principal de administrador	Haber accedido al juego previamente con rol 'Profesor'.	El menú principal del jugador debe mostrar las siguientes opciones: gestionar preguntas de administrador, gestionar preguntas de alumnos, visualizar rankings, gestionar peticiones, gestionar tableros de juego y acceder al juego.
PA - 10	Formato y funcionalidad de pregunta de juego	Hacer click en la ficha de juego situada en alguna casilla del tablero de tipo 'Pregunta por tema'.	Se debe mostrar una pregunta del tablero. Una vez hecho esto, se deben marcar las respuestas probando a cambiar la elección antes que finalice el tiempo. Se debe comprobar que la pregunta cuenta con un límite de tiempo (Gráfico y numérico), la dificultad (Por medio de estrellas), el enunciado, vídeo (opcional), las respuestas posibles y la opción de corregir la respuesta.

Answer2Pass – Juego de evaluación interactiva de alumnos

Identificador	Nombre	Entrada Esperada	Salida Esperada
PA - 11	Responder pregunta	Se debe haber mostrado una pregunta previamente y se debe responder a la misma.	El sistema devuelve el mensaje: <i>"Correct answers!"</i> si la respuesta es correcta o el mensaje: <i>"Sorry, wrong answers"</i> si es incorrecta. Según la amplitud de la tirada, el sistema representa la nueva posición en el tablero. Por ejemplo, si se responde de forma inmediata, la amplitud será 6, por lo que la ficha nueva debe disponerse a 6 casillas de la posición anterior. Por otro lado, si la respuesta es incorrecta, la amplitud será 1, por lo que la nueva ficha debe disponerse a una casilla de la posición anterior. El sistema añade una tirada más al marcador. El sistema añade al marcador el tiempo empleado en responder.
PA - 12	Funcionalidad de comodín: Tema aleatorio	Caer previamente en una casilla de tipo 'Comodín de tema aleatorio'. Hacer click sobre ella para desplegar la pregunta. Repetir este proceso varias veces.	Comprobar que no siempre se muestra el mismo tema de pregunta.
PA - 13	Funcionalidad de comodín: Comodín sorpresa	Caer previamente en una casilla de tipo 'Comodín sorpresa'. Hacer click sobre ella para desplegar la pregunta. Repetir este proceso varias veces.	Comprobar que se activan comodines diferentes.
PA - 14	Funcionalidad de comodín: Aumento de tirada	Caer previamente en una casilla de tipo 'Comodín de aumento de tirada'. Hacer click sobre ella para desplegar la pregunta.	Comprobar que, al responder la pregunta, se suma 2 a la amplitud de la tirada. Por ejemplo, si la pregunta se responde inmediatamente (amplitud 6), las nuevas posibles posiciones deben disponerse a una distancia de 8 de la casilla anterior.
PA - 15	Funcionalidad de comodín: Doble tiempo	Caer previamente en una casilla de tipo 'Comodín de doble tiempo'. Hacer click sobre ella para desplegar la pregunta.	El tiempo de respuesta se duplica.

Answer2Pass – Juego de evaluación interactiva de alumnos

Identificador	Nombre	Entrada Esperada	Salida Esperada
PA - 16	Funcionalidad de comodín: Envío de pregunta	Caer previamente en una casilla de tipo 'Comodín de nueva pregunta. Hacer click sobre ella para desplegar la pregunta.	Aparece un editor de preguntas para enviar a otro usuario. Este editor contiene los siguientes campos: tipo, enunciado, dificultad, respuestas y tiempo.
PA - 17	Tipos de pregunta	Jugar una partida previamente	Las preguntas respondidas son de distinto tema (según el color de la casilla) y de distinto tipo (True/False, una respuesta y varias respuestas)
PA - 18	Responder preguntas de otros usuarios: Responder de forma errónea	Acceder previamente al apartado de preguntas sin responder de otro usuario. Hacer click en alguna de ellas para mostrarla. Responder de forma errónea.	Puntuación de 'Mejor respondedor' no incrementada en el ranking. No se añade una nueva ficha de bonus al contenedor habilitado para su almacenamiento.
PA - 19	Responder preguntas de otros usuarios: Responder de forma correcta	Acceder previamente al apartado de preguntas sin responder de otro usuario. Hacer click en alguna de ellas para mostrarla. Responder de forma correcta.	Puntuación incrementada según la dificultad de la pregunta. Se añade una nueva ficha de bonus al contenedor habilitado para su almacenamiento.
PA - 20	Uso de ficha bonus	Haber activado una ficha bonus lograda al responder de forma correcta una pregunta recibida por un alumno.	Nueva posición en el tablero calculada a partir de una tirada de amplitud 2. Las nuevas posiciones posibles distan 2 casillas de la anterior.

Answer2Pass – Juego de evaluación interactiva de alumnos

Identificador	Nombre	Entrada Esperada	Salida Esperada
PA - 21	Pantalla de fin de partida y cálculo de puntuaciones	Completar partida previamente.	Se muestra una pantalla final que indica la puntuación final acumulada que se ha ido mostrando en el marcador. Se muestra el tiempo total de respuesta y el número de tiradas totales. Dichas puntuaciones deben reflejarse tal cuál en el ranking que almacena las puntuaciones de la última partida. Si la puntuación obtenida es mejor que alguna anterior, debe quedar reflejada en el ranking que muestra la mejor puntuación. El ranking que almacena la puntuación media, debe mostrar el tiempo y tiradas medias que se han obtenido incluyendo esta última partida.
PA - 22	Publicación de puntuación	Completar partida, visualizar pantalla de puntuaciones descrita en la prueba PA-21 y enviar puntuación al muro de Facebook de forma previa.	La puntuación obtenida en el juego (supóngase 4 tiradas en 10 segundos) se encuentra publicada en el muro de Facebook mostrando el siguiente mensaje: <i>"The score obtained in Answer2Pass was: 4 throws in 00:10"</i>
PA - 23	Nueva partida	Se inicia una nueva partida haciendo uso de la opción habilitada en el menú de la pantalla de juego.	Las puntuaciones del marcador se establecen en 0. La ficha de juego se encuentra situada en la posición inicial del tablero.
PA - 24	Creación de preguntas de juego	Acceder previamente a la pantalla de administrador que muestra las preguntas de juego creadas por los mismos. Haber seleccionado la opción de crear nueva pregunta habilitada.	Se muestra un creador de preguntas cuyos campos a rellenar son: tipo, tema, enunciado, vídeo (opcional), dificultad, respuestas correctas, respuestas posibles y tiempo.
PA - 25	Contenedor de preguntas de juego	Acceder previamente como administrador y crear una nueva pregunta de juego.	La pregunta se almacena en un contenedor de preguntas de juego

Answer2Pass – Juego de evaluación interactiva de alumnos

Identificador	Nombre	Entrada Esperada	Salida Esperada
PA - 26	Contenedores de preguntas enviadas por alumnos	Acceder previamente a la pantalla de administrador que muestra las preguntas enviadas por alumnos.	Las preguntas se disponen en dos contenedores: Por un lado se encuentran las que están sin corregir, y por otro las enviadas.
PA - 27	Campos de las preguntas en los contenedores	Haber accedido previamente al contenedor de preguntas de juego creadas, al contenedor de preguntas de alumnos sin corregir y al contenedor de preguntas de alumnos enviadas.	Cada pregunta del contenedor muestra los siguientes campos: identificador de la pregunta, el usuario de origen, el usuario de destino, el tema, el tiempo, la dificultad, la URL del vídeo (opcional), el enunciado y las respuestas.
PA - 28	Corregir y enviar pregunta	Haber accedido previamente al contenedor de preguntas sin corregir enviadas por los alumnos. Corregir dicha pregunta.	El alumno destinatario contiene la pregunta en el apartado de preguntas sin contestar enviadas por otros alumnos. La puntuación 'Mejor preguntador' del alumno que envió la pregunta aumenta en el ranking según la dificultad de la misma.
PA - 29	Notificar nueva pregunta enviada	Haber accedido previamente como administrador al contenedor de preguntas enviadas por alumnos. Corregir pregunta de alumno y enviar notificación de aviso.	El alumno destinatario recibe una notificación en el Facebook indicando el siguiente mensaje: <i>"Someone has sent you a question"</i> .

Answer2Pass – Juego de evaluación interactiva de alumnos

Identificador	Nombre	Entrada Esperada	Salida Esperada
PA - 30	Eliminar pregunta	Haber accedido previamente como administrador al contenedor de preguntas de juego creadas o al contenedor de preguntas enviadas por alumnos. Eliminar una pregunta del primer contenedor mencionado y otra del otro contenedor.	La pregunta se elimina del contenedor.
PA - 31	Lista de peticiones	Enviar una petición de forma previa y acceder como administrador a la lista de peticiones.	La petición aparece en una lista de peticiones pendientes mostrando los campos: nombre completo, el NIA, grupo del alumno, aceptar petición, rechazar petición, campo de entrada de texto, y enviar notificación.
PA - 32	Aceptar petición	Enviar una petición de forma previa y acceder como administrador a la lista de peticiones. Aceptar la petición.	La petición aparece en la lista de alumnos aceptados mostrando los campos: nombre completo, NIA y grupo.
PA - 33	Rechazar petición	Enviar una petición de forma previa y acceder como administrador a la lista de peticiones. Rechazar la petición.	La petición se elimina de la lista de peticiones.

Answer2Pass – Juego de evaluación interactiva de alumnos

Identificador	Nombre	Entrada Esperada	Salida Esperada
PA - 34	Notificar mensaje de petición	Enviar una petición de forma previa y acceder como administrador a la lista de peticiones. Redactar mensaje en el espacio habilitado para ello y enviar la notificación.	La notificación se recibe en el Facebook del alumno que ha enviado la petición enunciando el mensaje escrito por el administrador.
PA - 35	Generar tablero con colores y comodines dispuestos de forma aleatoria	Acceder como administrador y rellenar el nombre del tablero, el nombre de los temas, y marcar los colores que aparecerán.	Aparece en una nueva ventana el tablero generado. Las casillas de colores y comodines se disponen sin seguir ningún patrón.
PA - 36	Lista de tableros generados	Acceder como administrador y generar un nuevo tablero de juego.	Se ha añadido un nuevo tablero a la lista, mostrando los campos: nombre del tablero, temas que trata e iconos de visualización y borrado.
PA - 37	Visualizar un tablero	Acceder como administrador a la lista de tableros creados. Visualizar un tablero cualquiera.	El tablero se visualiza en una nueva ventana. El nombre del tablero se corresponde, así como el número de casillas de distinto color con el número de temas de los que trata.
PA - 38	Borrar un tablero	Acceder como administrador a la lista de tableros creados. Borrar un tablero cualquiera.	El tablero se borra de la lista.
PA - 39	Seleccionar un tablero	Acceder como administrador a la lista de tableros creados. Visualizar un tablero cualquiera, seleccionarlo y guardar los cambios.	Comprobar que ese mismo tablero es el que se visualiza en la pantalla de juego.

Answer2Pass – Juego de evaluación interactiva de alumnos

Identificador	Nombre	Entrada Esperada	Salida Esperada
PA - 40	Filtrar preguntas	Acceder como administrador a cualquier contenedor de preguntas. Introducir el tema de pregunta y filtrar las mismas. Introducir el tipo de pregunta y filtrar las mismas. Introducir la dificultad de pregunta y filtrar las mismas.	Sólo se visualizan las preguntas que tratan del tema introducido. Sólo se visualizan las preguntas que son del tipo seleccionado. Sólo se visualizan las preguntas que poseen la dificultad introducida.

Pruebas 1: Catálogo de pruebas

2.8.3. Matriz Requisitos funcionales – Pruebas

	PA - 01	PA - 02	PA - 03	PA - 04	PA - 05	PA - 06	PA - 07	PA - 08	PA - 10	PA - 11	PA - 12	PA - 13	PA - 14	PA - 15	PA - 16	PA - 17	PA - 18	PA - 19	PA - 20	PA - 21	PA - 22	PA - 23	PA - 24	PA - 25	PA - 26	PA - 27	PA - 28	PA - 29	PA - 30	PA - 31	PA - 32	PA - 33	PA - 34	PA - 35	PA - 36	PA - 37	PA - 38	PA - 39	PA - 40	
RF - 01		X																																						
RF - 02					X																																			
RF - 03			X	X	X																																			
RF - 04						X																																		
RF - 05																		X																						

Answer2Pass – Juego de evaluación interactiva de alumnos

	PA - 01	PA - 02	PA - 03	PA - 04	PA - 05	PA - 06	PA - 07	PA - 08	PA - 09	PA - 10	PA - 11	PA - 12	PA - 13	PA - 14	PA - 15	PA - 16	PA - 17	PA - 18	PA - 19	PA - 20	PA - 21	PA - 22	PA - 23	PA - 24	PA - 25	PA - 26	PA - 27	PA - 28	PA - 29	PA - 30	PA - 31	PA - 32	PA - 33	PA - 34	PA - 35	PA - 36	PA - 37	PA - 38	PA - 39	PA - 40		
RF - 06																		X	X																							
RF - 07										X																																
RF - 08											X																															
RF - 09																		X	X																							
RF - 10																		X	X																							
RF - 11												X	X	X	X	X																										
RF - 12																						X																				
RF - 13																					X																					
RF - 14																							X																			
RF - 15											X																															
RF - 16											X																															
RF - 17											X																															
RF - 18										X																																

Answer2Pass – Juego de evaluación interactiva de alumnos

	PA - 01	PA - 02	PA - 03	PA - 04	PA - 05	PA - 06	PA - 07	PA - 08	PA - 10	PA - 11	PA - 12	PA - 13	PA - 14	PA - 15	PA - 16	PA - 17	PA - 18	PA - 19	PA - 20	PA - 21	PA - 22	PA - 23	PA - 24	PA - 25	PA - 26	PA - 27	PA - 28	PA - 29	PA - 30	PA - 31	PA - 32	PA - 33	PA - 34	PA - 35	PA - 36	PA - 37	PA - 38	PA - 39	PA - 40	
RF - 19															X																									
RF - 20									X																															
RF - 21																							X																	
RF - 22																								X																
RF - 23																									X															
RF - 24																													X											
RF - 25																												X												
RF - 26																														X										
RF - 27																																X	X	X						
RF - 28																																			X					
RF - 29																																			X					
RF - 30																																					X			
RF - 31																																						X		

Answer2Pass – Juego de evaluación interactiva de alumnos

	PA - 01	PA - 02	PA - 03	PA - 04	PA - 05	PA - 06	PA - 07	PA - 08	PA - 10	PA - 11	PA - 12	PA - 13	PA - 14	PA - 15	PA - 16	PA - 17	PA - 18	PA - 19	PA - 20	PA - 21	PA - 22	PA - 23	PA - 24	PA - 25	PA - 26	PA - 27	PA - 28	PA - 29	PA - 30	PA - 31	PA - 32	PA - 33	PA - 34	PA - 35	PA - 36	PA - 37	PA - 38	PA - 39	PA - 40	
RF - 32																																						X		
RF - 33	X																																							
RF - 34	X																																							
RF - 35										X																														
RF - 36							X																																	
RF - 37								X																																
RF - 38						X																																		
RF - 39									X																															
RF - 41																				X																				
RF - 42																X																								
RF - 43														X																										
RF - 44																							X																	
RF - 45																										X														

Answer2Pass – Juego de evaluación interactiva de alumnos

	PA - 01	PA - 02	PA - 03	PA - 04	PA - 05	PA - 06	PA - 07	PA - 08	PA - 10	PA - 11	PA - 12	PA - 13	PA - 14	PA - 15	PA - 16	PA - 17	PA - 18	PA - 19	PA - 20	PA - 21	PA - 22	PA - 23	PA - 24	PA - 25	PA - 26	PA - 27	PA - 28	PA - 29	PA - 30	PA - 31	PA - 32	PA - 33	PA - 34	PA - 35	PA - 36	PA - 37	PA - 38	PA - 39	PA - 40		
RF - 46																															X										
RF - 47																																X									
RF - 48																																			X						
RF - 49																																				X					
RF - 50																																									X

Tabla 4: Matriz Requisitos - Pruebas

3. Diseño

En este capítulo se van a distinguir tres apartados diferenciados. En primer lugar se explica de forma detallada la arquitectura del sistema, que modelo se ha seguido para su desarrollo y los componentes que lo forma. Posteriormente, a más bajo nivel, se detallan todas las clases y métodos que forman las diferentes partes de la aplicación así como el diseño de todas las interfaces con las que el usuario puede interactuar. Por último, se presentarán algunos diagramas de secuencia que detallarán algunas acciones relevantes a realizar en la aplicación. Cabe destacar que en diversos apartados de este punto se hace referencia al profesor como administrador y viceversa, aclarar que aunque se nombren en ocasiones los dos términos poseen el mismo significado para la aplicación.

3.1. Arquitectura del sistema

A continuación se explicará detalladamente la arquitectura del sistema a desarrollar. Se establecerá que tipo de aplicación se está implementando, el modelo que sigue y los componentes que la forman.

3.1.1. Tipo de aplicación a desarrollar

La aplicación a desarrollar en este proyecto tendrá una arquitectura de tipo cliente-servidor, por lo que los clientes accederán haciendo uso de un navegador e intercambiarán datos con un servidor, que se encargará de responder todas las peticiones recibidas. Como se puede ver en la figura 20, la aplicación será ejecutada gracias a tres partes diferenciadas que intercambiarán información en todo momento. Estas serán las máquinas clientes (Usuarios de la aplicación) y por parte del servidor, Facebook y Google App-Engine.

En cuanto a la primera parte, **los usuarios de la aplicación**, a través de un navegador web accederán primero a su cuenta de Facebook y posteriormente, ya sea a través de una notificación recibida o usando la búsqueda de aplicaciones y contactos que ofrece Facebook, accederán a ver el marco donde la red social cargará el contenido de la aplicación. Una vez hecho esto el usuario ya puede interactuar con la aplicación, que estará alojada en Google App-Engine y que a través de la ejecución de determinadas funciones, intercambiará datos con Facebook.

La segunda parte, **Facebook**, será la encargada de mostrar al usuario todo el contenido de la aplicación web, permitiendo a su vez acceder a partes de la red social

Answer2Pass – Juego de evaluación interactiva de alumnos

como el chat, la barra de herramientas superior, las páginas recomendadas, personas que quizá conozcas etc. Como se ha mencionado anteriormente, la aplicación estará alojada en Google App-Engine, por lo que la red social accederá a su contenido y lo cargará en el espacio reservado.

La tercera parte, **Google App-Engine**, será la parte del servidor ocupado en alojar toda la aplicación (Código fuente, imágenes, librerías etc.). Este contenido será solicitado por Facebook para poder mostrarlo al usuario. A su vez, el código contendrá funciones que realizarán peticiones a la red social para intercambiar datos. Dicho intercambio se explicará de forma mucho más detallada en el apartado 4.1.1. propio de la implementación. Un ejemplo de ello sería el envío de notificaciones y las publicaciones en el muro llevadas a cabo a través de la aplicación y la obtención de datos como la imagen, email, el nombre o el identificador del usuario.

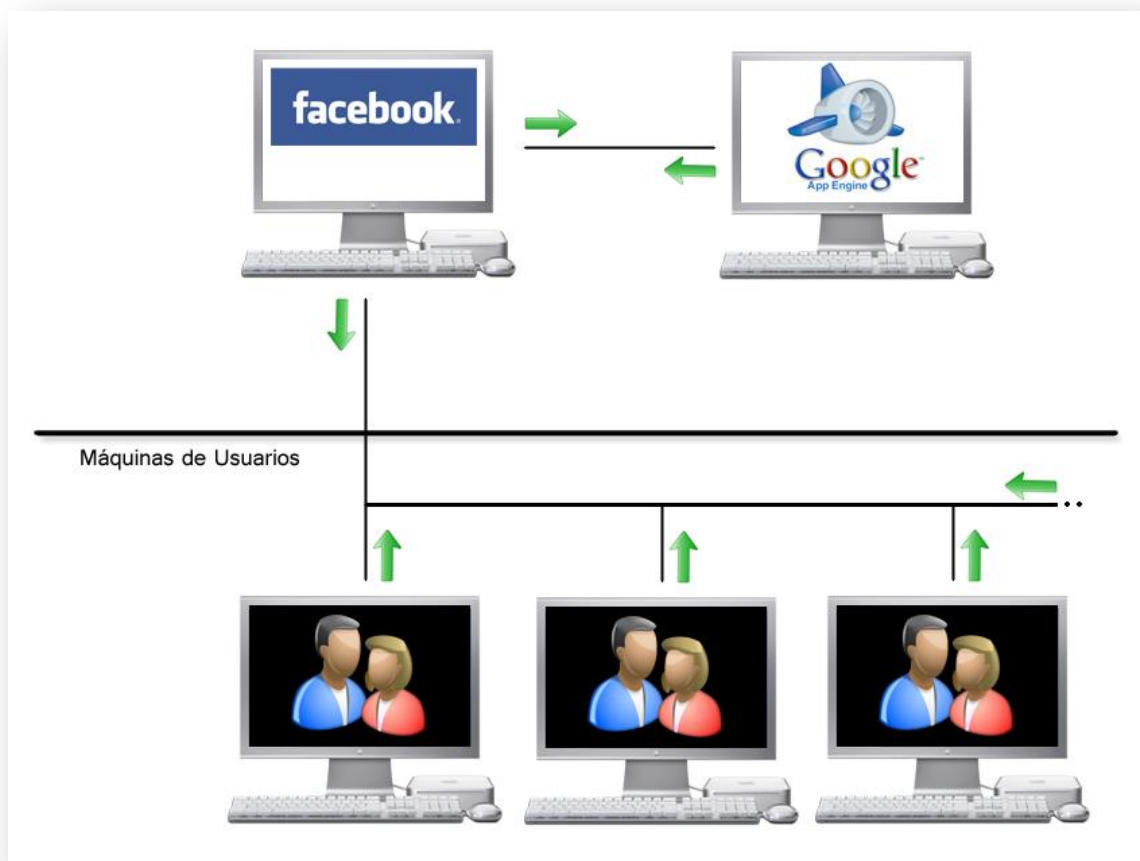


Figura 20: Conexión Cliente - Facebook – Google App Engine

3.1.2. Modelo a seguir

El modelo seguido para realizar el diseño de la arquitectura del sistema será el *Model-View-Controller*(MVC). Como su propio nombre indica, dicho modelo divide la interacción entre el usuario y la aplicación en tres roles, el modelo (Lógica de negocio), la vista (Interfaz de usuario) y el controlador. El principal objetivo de este modelo es facilitar el crecimiento por separado de cada una de las partes, incrementar así la reutilización y flexibilidad y facilitar el desarrollo independiente, las pruebas y el mantenimiento de cada rol.

Este modelo fue descrito por primera vez en 1979 por Trygve Reens Kaug, un informático profesor de la universidad de Oslo, cuando trabajaba en laboratorios de Xerox en SmallTalk, lenguaje por el cual un entorno de objetos interacciona a través del envío de mensajes con un mundo virtual.

La función que desempeña cada rol de este modelo es la siguiente:

- **Modelo:** Es la parte de la arquitectura de la aplicación que representa la información con la que opera el sistema. El modelo responde a las peticiones de información, principalmente desde la vista, y a las instrucciones recibidas para realizar cambios, generalmente desde el controlador. Estos cambios de notifican a los usuarios a través de la vista.
- **Vista:** Es la parte de la arquitectura de la aplicación encargada de presentar el modelo al usuario de forma adecuada para la interacción, generalmente en forma de interfaz. Pueden existir múltiples vistas para cubrir varios propósitos.
- **Controlador:** Parte de la arquitectura de la aplicación encargada de recibir, y responder los eventos que vienen de parte del usuario. Se ocupa del tratamiento de datos y realiza peticiones tanto a la vista como al modelo.

El proceso que sigue esta arquitectura es el siguiente:

- El usuario interactúa de alguna forma con la parte de la vista (Interfaz), por ejemplo, pulsando un botón encargado de enviar un formulario.
- El controlador recibe el evento enviado por la vista y lo gestiona.
- Posteriormente, el controlador accede al modelo y este se actualiza. Por ejemplo, en este caso almacena los datos del formulario.
- La vista obtiene los datos que el modelo actualizado manda pasando por el controlador (Si requiere tratamiento de datos), actualizando así la interfaz notificando los cambios producidos. En este caso, por ejemplo se mostraría al usuario por pantalla los datos que envió pulsando el botón.
- La interfaz espera de nuevo el envío de eventos por parte del usuario de la aplicación.

Cabe destacar que en algunas ocasiones la vista puede acceder directamente al modelo sin pasar previamente por el controlador. Esto sucede cuando no se requiere ningún tipo de tratamiento de datos y se desea acceder directamente a la base de almacenamiento de los mismos.

Todo este ciclo se puede observar en la siguiente imagen:

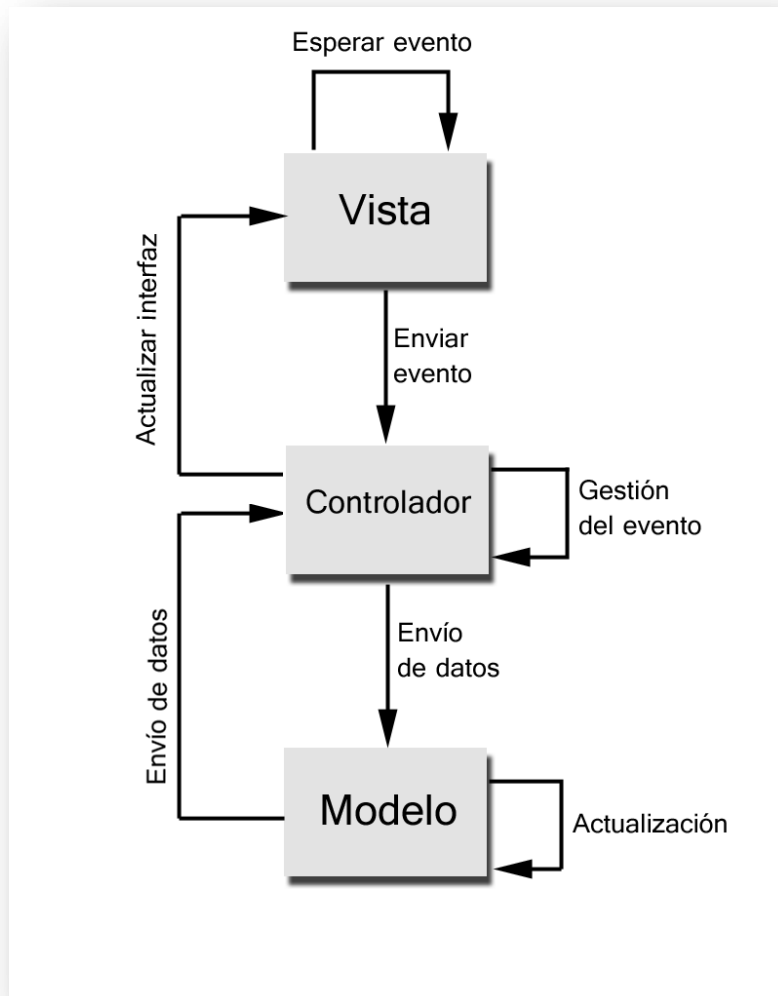


Figura 21: Modelo Vista Controlador

3.1.3. Diagrama de componentes

A continuación se detalla el diagrama que representa los componentes que forman la aplicación a más alto nivel. Este se encuentra dividido principalmente en tres partes: La parte correspondiente a la **vista**, formada por todas las interfaces con las que los usuarios pueden interactuar; La parte dedicada al **controlador** de la aplicación, formada por aquellos componentes encargados de gestionar las preguntas, gestionar las peticiones y usuarios, gestionar el ranking y gestionar la partida; La tercera parte es la que representa el **modelo**, o lo que es lo mismo, la parte encargada de almacenar y proporcionar todos los datos necesarios para el funcionamiento de la aplicación. Esta última parte contiene componentes como los referidos a los datos de las preguntas, los datos propios de cada usuario, los datos de las peticiones y los datos de la partida. Por último, se diferencia otro componente aislado de las tres partes del MVC, el dedicado a las **librerías externas de Facebook**. Este necesitará componentes como el de gestión usuarios o el de gestión de partidas para poder llevar a cabo las funcionalidades relacionadas con la red social. El diagrama que muestra las relaciones entre todos los componentes es el siguiente:

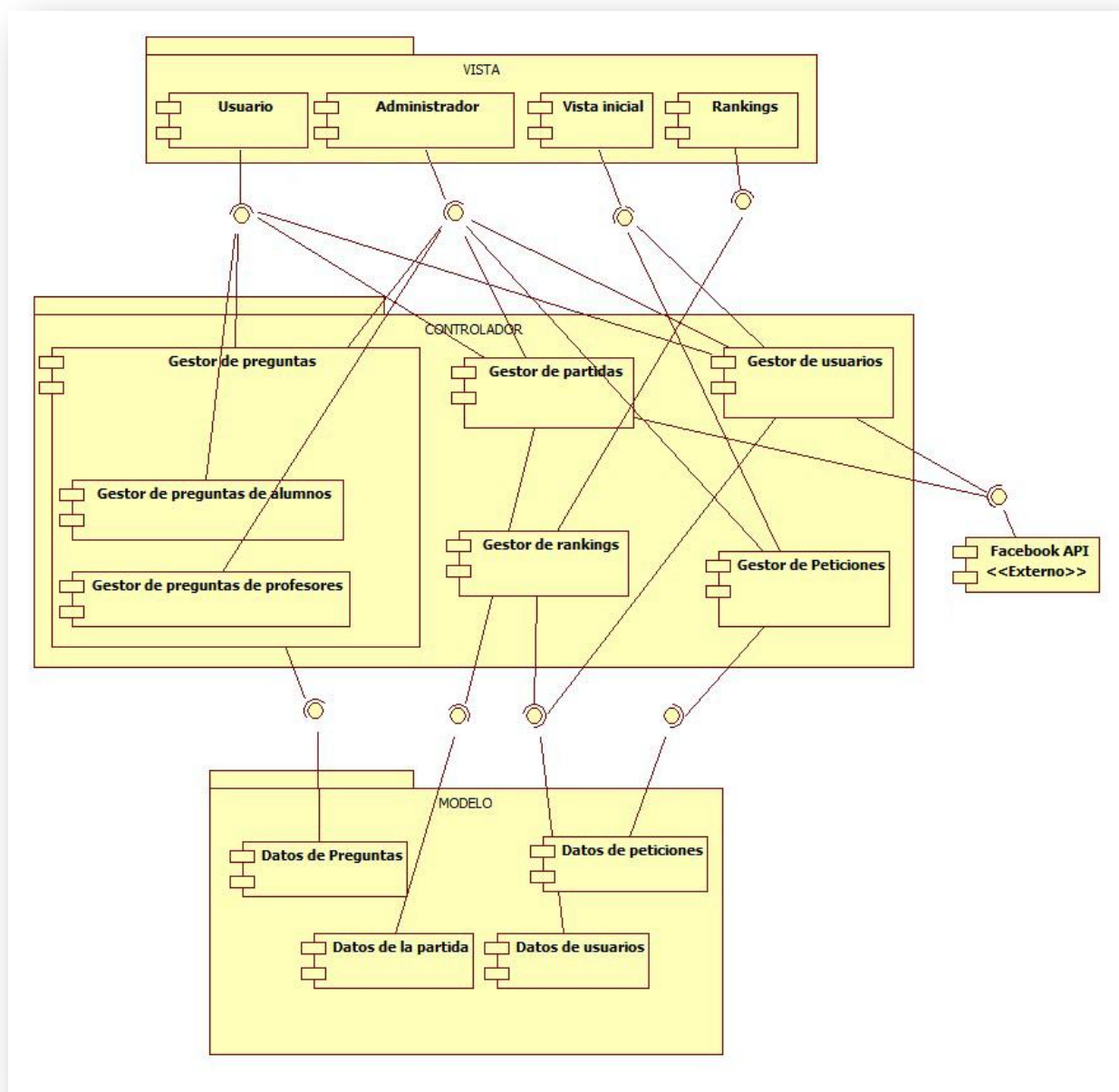


Figura 22: Componentes de la aplicación

3.2 Diseño detallado

A continuación se detallan todas las clases y métodos que forman las diferentes partes de la aplicación, obteniendo así el diseño de la misma a más bajo nivel. Los roles correspondientes al MVC en dicha aplicación se identifican de la siguiente forma:

- **Vista:** Formado por todos los ficheros JSP de la aplicación, que a su vez incluirán todas las librerías, imágenes y ficheros de estilo. Todo ello dará lugar a las interfaces con las que el usuario podrá interactuar (Pulsando botones, introduciendo datos etc.). Como se ha mencionado en el punto 2.1.2, la vista se encontrará todo el tiempo a la espera de recibir eventos por parte del usuario, una vez ocurre esto se ocupa de transmitirlos al controlador y al modelo.
- **Controlador:** El controlador de la aplicación estará formado por toda la estructura de *servlets* creada. Los *servlets* son un tipo de tecnología web servidor implementados como clases Java que se ocupan de recibir peticiones de usuarios, de gestionarlas en el servidor y de devolver la respuesta del mismo. De esta forma, se encargarán de recibir los eventos enviados por los usuarios a través de la vista, gestionarán los datos recibidos aplicando métodos determinados para cada caso y por último, accederán al modelo para actualizar los datos.
- **Modelo:** El modelo de la aplicación a desarrollar estará formado por todas las clases Java diseñadas para crear objetos que almacenarán datos gracias a la tecnología JDO. De esta forma, el modelo obtendrá por parte del controlador los datos, y se ocupará de instanciar objetos por los cuáles se actualizará la base de datos de la aplicación. Una vez hecho esto, cada vez que el usuario (A través de la vista), vuelva a acceder a dichos datos, podrá ver los cambios producidos.

A continuación se puede observar un diagrama que identifica todas las clases de la aplicación junto con sus métodos y atributos correspondientes, indicando mediante relaciones las clases de las que dependen. Estas clases han sido agrupadas por bloques de colores que ayudan a diferenciar de forma clara todos los roles del modelo MVC empleado para diseñar la arquitectura de la aplicación:

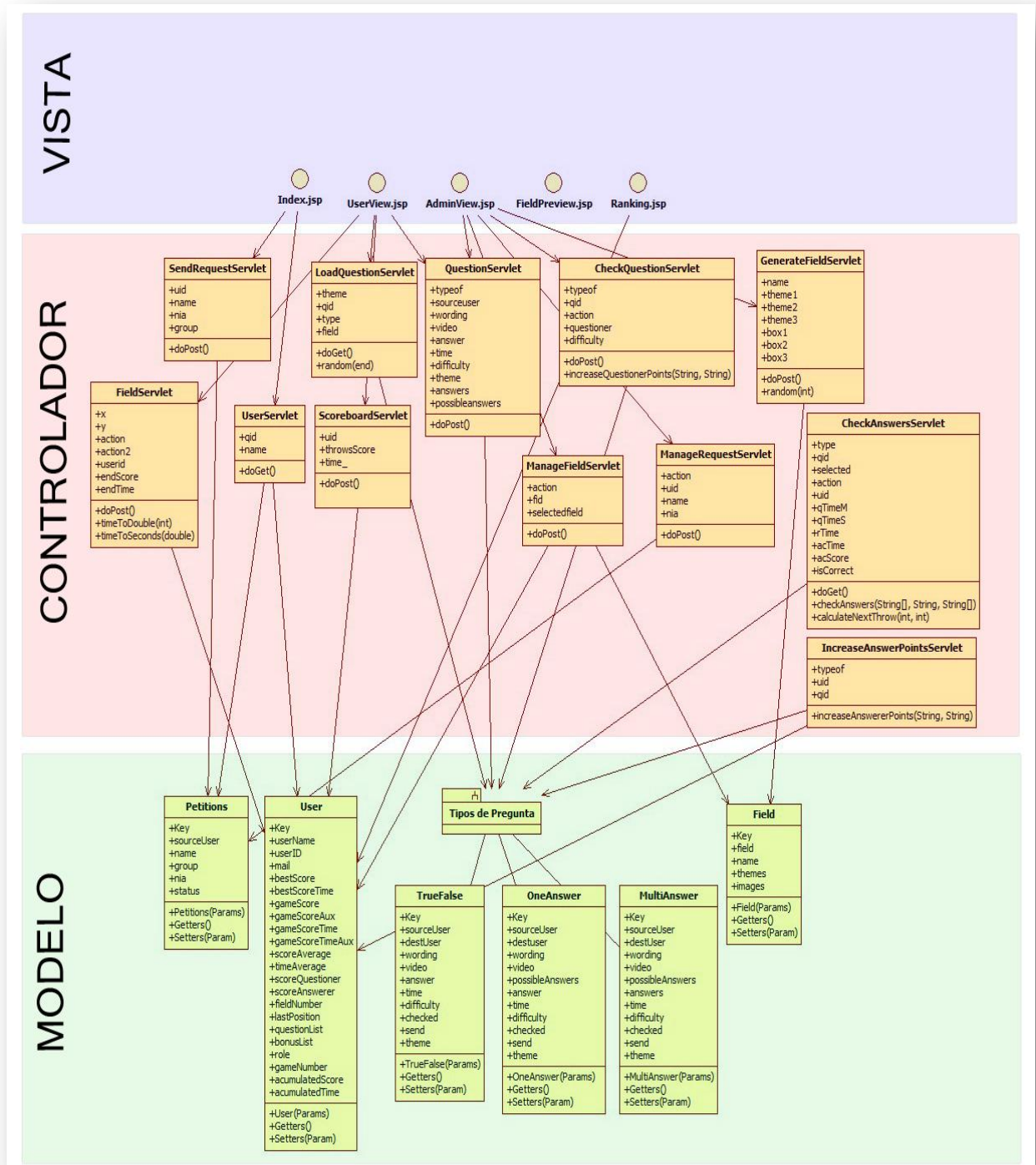


Figura 23: Relaciones existentes entre las clases de la aplicación

Todas las clases establecidas en el diagrama anterior se encuentran incluidas dentro de los componentes especificados en la figura 22. Según las diferentes partes que forman el MVC las relaciones entre clases y componentes son las siguientes:

- **Vista:** En cuanto a las interfaces que componen la aplicación, el archivo Index.jsp se incluye dentro del componente 'Vista Inicial', el archivo UserView.jsp dentro del componente 'Usuario', el archivo Rankings.jsp dentro de 'Rankings' y por último, los ficheros FieldPreview.Jsp y AdminView.jsp dentro del componente 'Administrador'.
- **Controlador:** Para comenzar, dentro del componente 'Gestor de preguntas' se incluirán los controladores 'QuestionServlet' y 'CheckQuestionServlet', que serán empleados para gestionar tanto las preguntas de alumnos como las de profesores. Por otro lado, el componente 'Gestor de Usuarios' contiene el controlador 'UserServlet', encargado de visualizar los datos de los usuarios registrados, su rol etc. El componente 'Gestor de partidas', incluirá los controladores 'ManageFieldServlet', 'GenerateFieldServlet', 'LoadQuestionServlet', 'CheckAnswersServlet', 'FieldServlet' y 'ScoreBoardServlet', todos ellos encargados de gestionar y crear los tableros de juego, controlar las puntuaciones mostradas en el marcador, cargar la pregunta de juego, comprobar la validez de las respuestas enviadas por los usuarios y realizar los cálculos pertinentes que permitan el movimiento de la ficha por el tablero. En cuanto a los rankings, el componente 'Gestión de Rankings' incluirá el controlador 'IncreaseAnswerPointsServlet', encargado de aumentar la puntuación de los jugadores que respondan de forma correcta las preguntas enviadas por otros alumnos. Finalmente, el componente 'Gestión de Peticiones' contiene los controladores 'ManageRequestServlet' y 'SendRequestServlet', encargados de gestionar las peticiones recibidas y el propio envío de estas respectivamente.
- **Modelo:** En la parte relacionada con el modelo de la arquitectura encontramos en primer lugar el componente denominado 'Datos de Preguntas', que incluirá todas aquellas clases encargadas de guardar datos relacionados con las preguntas creadas en la aplicación. Estas clases son 'TrueFalse', 'OneAnswer' y 'Multianswer', que se encargarán de almacenar todos los atributos de las preguntas según el tipo. Por otro lado, el componente 'Datos de partida' incluirá la clase 'Field', encargada de almacenar todos los datos referentes a los tableros de juego. La clase denominada 'Petitions' se encontrará dentro del componente 'Datos de peticiones'. Dicha clase será la encargada de almacenar todos los datos relacionados con las peticiones de acceso enviadas por los usuarios de la aplicación. Por último, el componente 'Datos de Usuarios' incluirá la clase denominada 'Users', que almacenará todos los datos propios de los usuarios registrados. Entre estos se encuentran el nombre completo, el identificador de Facebook, el email, las puntuaciones obtenidas, el tablero que juega, el rol etc.

Answer2Pass – Juego de evaluación interactiva de alumnos

A continuación se van a explicar de forma detallada todas y cada una de las clases presentes en cada uno de los tres bloques del modelo. Se explicará la funcionalidad de cada una de ellas, así como de sus métodos y atributos.

3.2.1. Diseño de la Interfaz

La interfaz con la que el usuario puede interactuar en la aplicación (Vista), y a través de la cual se comunica con el servidor, está formada por cinco ficheros con extensión *jsp*. Estos son:

- **Index:** Es la primera pantalla que el usuario puede ver al ejecutar la aplicación. Esto queda configurado de esta forma en el servidor añadiendo la siguiente línea al archivo *web.xml*:

```
<welcome-file-list>
    <welcome-file>index.jsp</welcome-file>
</welcome-file-list>
```

En esta pantalla el usuario puede observar el logo y el lema de la aplicación, el creador y el logo de la universidad, una serie de pasos necesarios para poder acceder al contenido y un formulario disponible para todos los usuarios que quieran participar. Las acciones que el usuario puede realizar son:

- **Acceder al juego**, pulsando sobre el logo de la aplicación.
- **Enviar petición**, a través del formulario establecido pulsando el botón '*Send*' una vez que se han completado todos los campos.
- **Adquirir permisos**, pulsando en el botón implementado para dicha acción llamado '*Permissions*'.

La apariencia de esta primera interfaz es la siguiente:

Answer2Pass – Juego de evaluación interactiva de alumnos

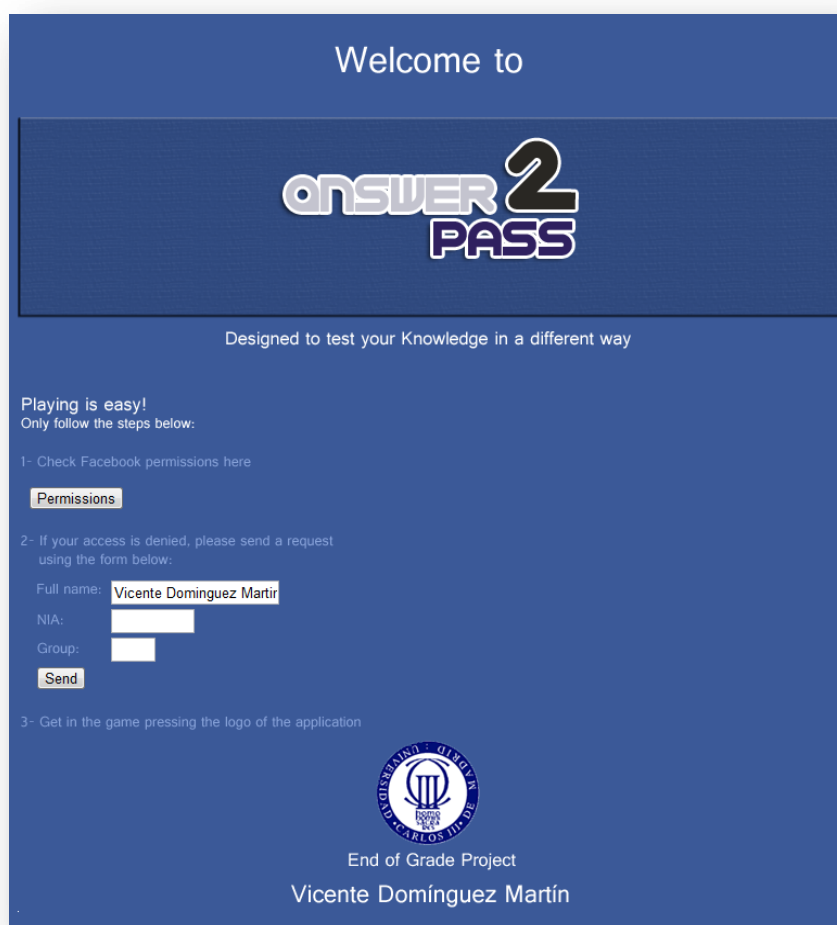


Figura 24: Interfaz de inicio, Answer2Pass

- **UserView:** Es la pantalla principal de juego, a la que sólo los usuarios cuya petición ha sido aceptada pueden acceder. En este fichero se ubica tanto la parte del tablero de juego como el contenedor de preguntas sin responder recibidas por otros usuarios.

En ella se puede observar un menú con diferentes opciones, el tablero de juego, el marcador, el contenedor de fichas bonus y el contenedor de preguntas sin responder. Cuando la partida finalice, también se podrá observar la pantalla encargada de mostrar las puntuaciones finales obtenidas.

Las acciones que un usuario puede realizar son:

- Hacer uso del **menú**, haciendo *click* en cada una de sus opciones (Nueva partida, rankings, preguntas sin responder recibidas por otros usuarios y tablero de juego).
- Mostrar una **pregunta enviada por otro usuario**, haciendo *click* en cualquiera de las disponibles en el contenedor.

Answer2Pass – Juego de evaluación interactiva de alumnos

- **Responder preguntas**, seleccionando la respuesta/s y haciendo *click* en el botón “*Check answers*”.
- **Crear pregunta**, cayendo en el comodín que permite esa funcionalidad.
- **Usar bonus**, haciendo *click* en cualquier ficha contenida en el “*Bonus Box*”.
- **Efectuar jugada**, haciendo click en la casilla donde se encuentre la ficha de juego (Incluye responder pregunta de juego).
- **Enviar puntuación**, haciendo uso del icono habilitado en la esquina inferior derecha de la pantalla de puntuación final.

La apariencia de esta interfaz es la siguiente:



Figura 25: Interfaz de juego I, Answer2Pass

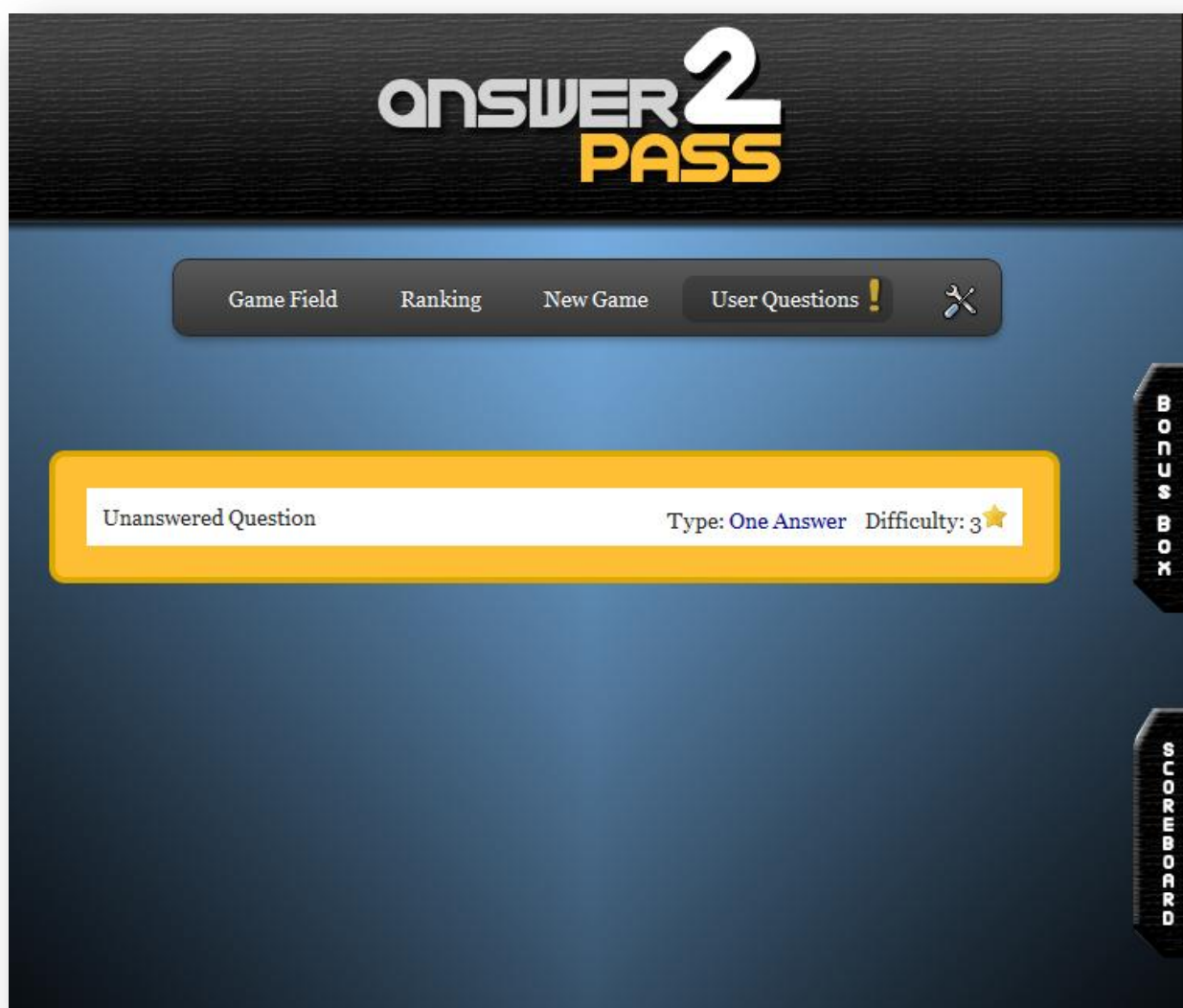


Figura 26: Interfaz de juego II, Answer2Pass

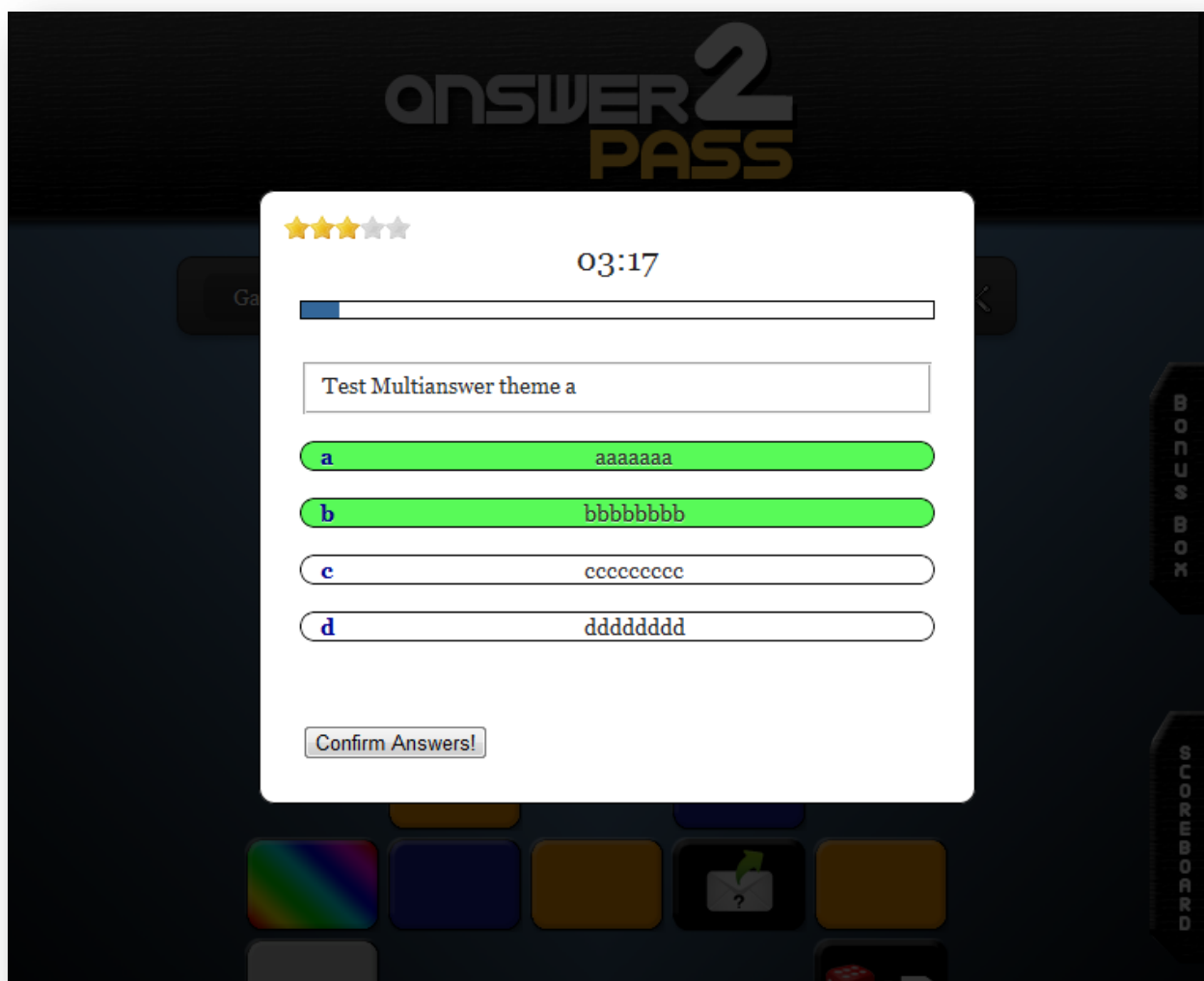


Figura 27: Interfaz de juego III, Answer2Pass

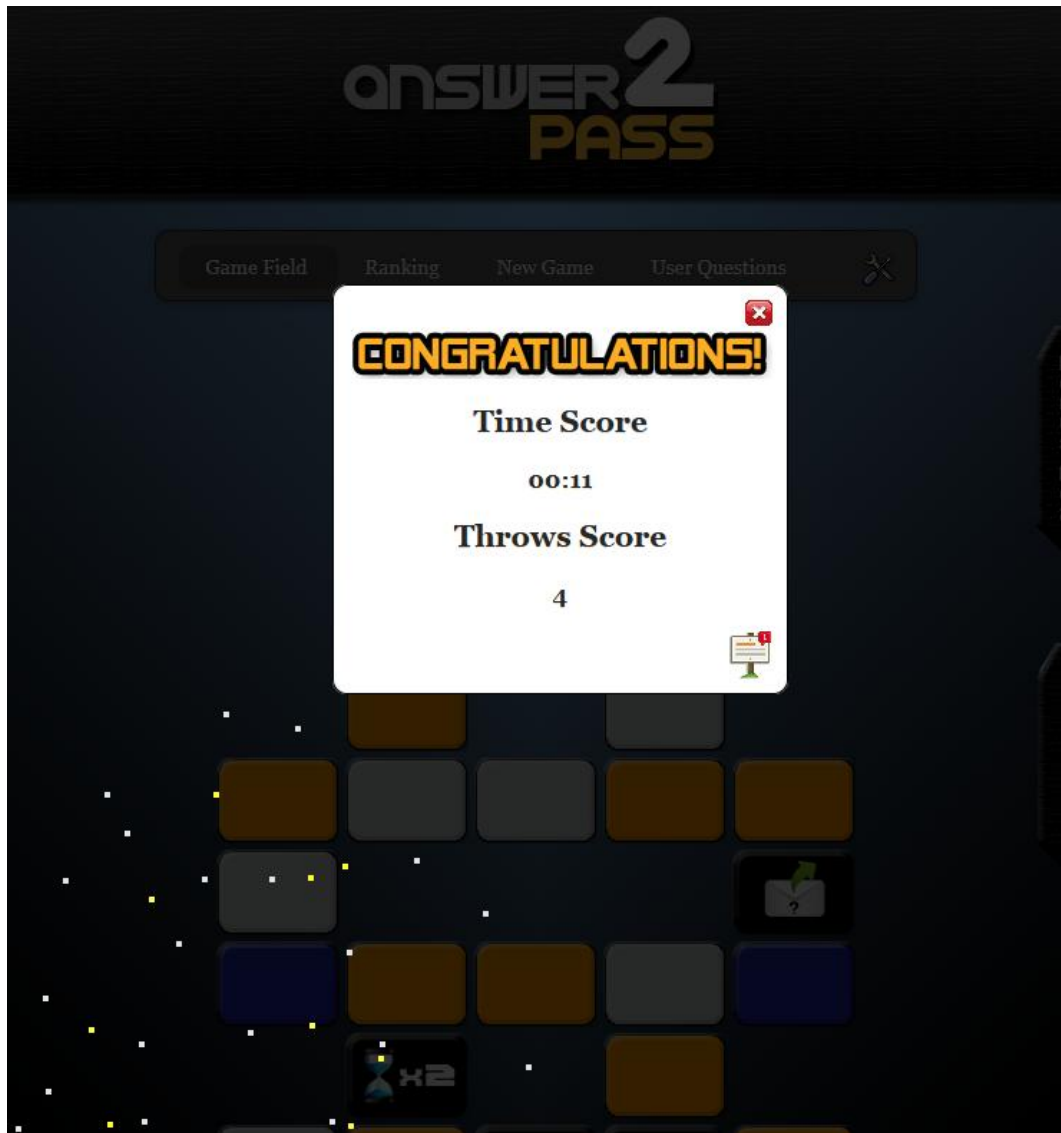


Figura 28: Interfaz de juego IV, Answer2Pass

- **AdminView:** Es la pantalla que visualizará el administrador del juego. A través de ella podrá crear todo tipo de preguntas para que los usuarios respondan en el juego y podrá corregir y enviar aquellas que se construyan por parte de los alumnos.

Esta interfaz será la encargada de recoger los parámetros introducidos por el administrador para crear nuevos tableros con casillas aleatorias, dando la posibilidad de visualizarlos y de establecerlos como tablero de juego. Por otro lado, esta interfaz permitirá al administrador gestionar todas las peticiones recibidas, aceptándolas o rechazándolas.

Las acciones que el administrador puede realizar son:

- Hacer uso del **menú**, haciendo click en cada una de sus opciones (Nueva pregunta, rankings, preguntas de alumnos, peticiones y generador de tableros).
- **Crear pregunta**, pulsando el botón “*New Question*”.
- **Corregir y enviar pregunta de alumno**, accediendo al contenedor de preguntas de alumnos y pulsando el botón “*Check and Send*”.
- **Filtrar preguntas, haciendo** uso de los filtros habilitados en cada contenedor de preguntas.
- **Enviar notificación**, pulsando el icono para enviar notificación en cada pregunta de alumno.
- **Generar nuevo tablero**, rellenando el formulario destinado para ello y pulsando el botón “*Generate*”.
- **Visualizar tablero creado**, haciendo *click* en la lupa situada en cada uno de los tableros creados de la lista.
- **Establecer nuevo tablero de juego**, seleccionando el tablero y pulsando el botón “*Save*”.
- **Eliminar pregunta**, pulsando el icono con forma de aspa roja establecido en la esquina superior derecha de cada pregunta.
- **Aceptar petición**, pulsando el icono situado en la columna “*Accept*” en la tabla de peticiones recibidas.
- **Rechazar petición**, pulsando el icono situado en la columna “*Reject*” en la tabla de peticiones recibidas.
- **Notificar aceptación o rechazo de la petición**, pulsando el icono situado en la columna “*Notify*” en la tabla de peticiones recibidas.

La apariencia de esta interfaz es la siguiente:

ANSWER2 PASS

Admin View

Admin Questions

Student Questions

Requests

Field Generator

Ranking

Question Container

Type: - Theme Difficulty: - Filter Show All

+?

True / False

Question Type: True / False

Question ID: 293

Source User: admin

Dest. User: admin

Difficulty: 2

Time: 2.0

Content:

Theme: b

Wording: Tema b TrueFalse

Video: URL

Image: NO IMAGE

Answer: True

Question Type: True / False

Question ID: 294

Figura 29: Interfaz de administración I, Answer2Pass


Admin Questions

Student Questions

Requests

Field Generator

Ranking



Question Container

Type: Theme Difficulty:

FilterShow All

True / False

Question Type: True / False

Question ID: 306

Source User: student

Dest. User: 100003473141931

Difficulty: 4

Time(min): 4.0

Content:


Theme: studentTheme

Wording: Treue False Question

Video: URL

Image: NO IMAGE

Answer: True

 Check & Send

Question Type: True / False

Question ID: 336

Source User: student

Dest. User: 100003473141931

Difficulty: 3

Time(min): 3.0

Content:


Theme: studentTheme

Wording: Pruebita True

Video: URL




Image: NO IMAGE

Figura 30: Interfaz de administración II, Answer2Pass

Admin Questions
Student Questions
Requests
Field Generator
Ranking


Field Generator

Field name:

Theme	Colour	Add
<input type="text"/>		<input type="checkbox"/>
<input type="text"/>		<input type="checkbox"/>
<input type="text"/>		<input type="checkbox"/>

Follow the steps below to generate fields correctly:

- 1- Write the name of the field
- 2- Write the name of each theme that you want to appear in the game
- 3- Add definitely the Colour - Theme relation to the field configuration
- 4- Generate it!

Figura 31: Interfaz de administración III, Answer2Pass

Field List





Select Field Name	Themes	Delete	Preview
<input type="radio"/> Test	[c][v]		
<input type="radio"/> Test2	[Theme1][Theme2][Theme3]		

Figura 32: Interfaz de administración IV, Answer2Pass

ANSWER² PASS

Admin View


Admin Questions

Student Questions



Requests!

Field Generator

Ranking



Pending Requests

Name	NIA	Group	Accept	Reject	Notify a message
Nuria Ruiz Rubio	100081596	73			<input type="text"/> 

Student List

Name	NIA	Group
Jose Maria de Fuentes	100039302	81
Jorge Blasco	999999	99

Figura 33: Interfaz de administración V, Answer2Pass

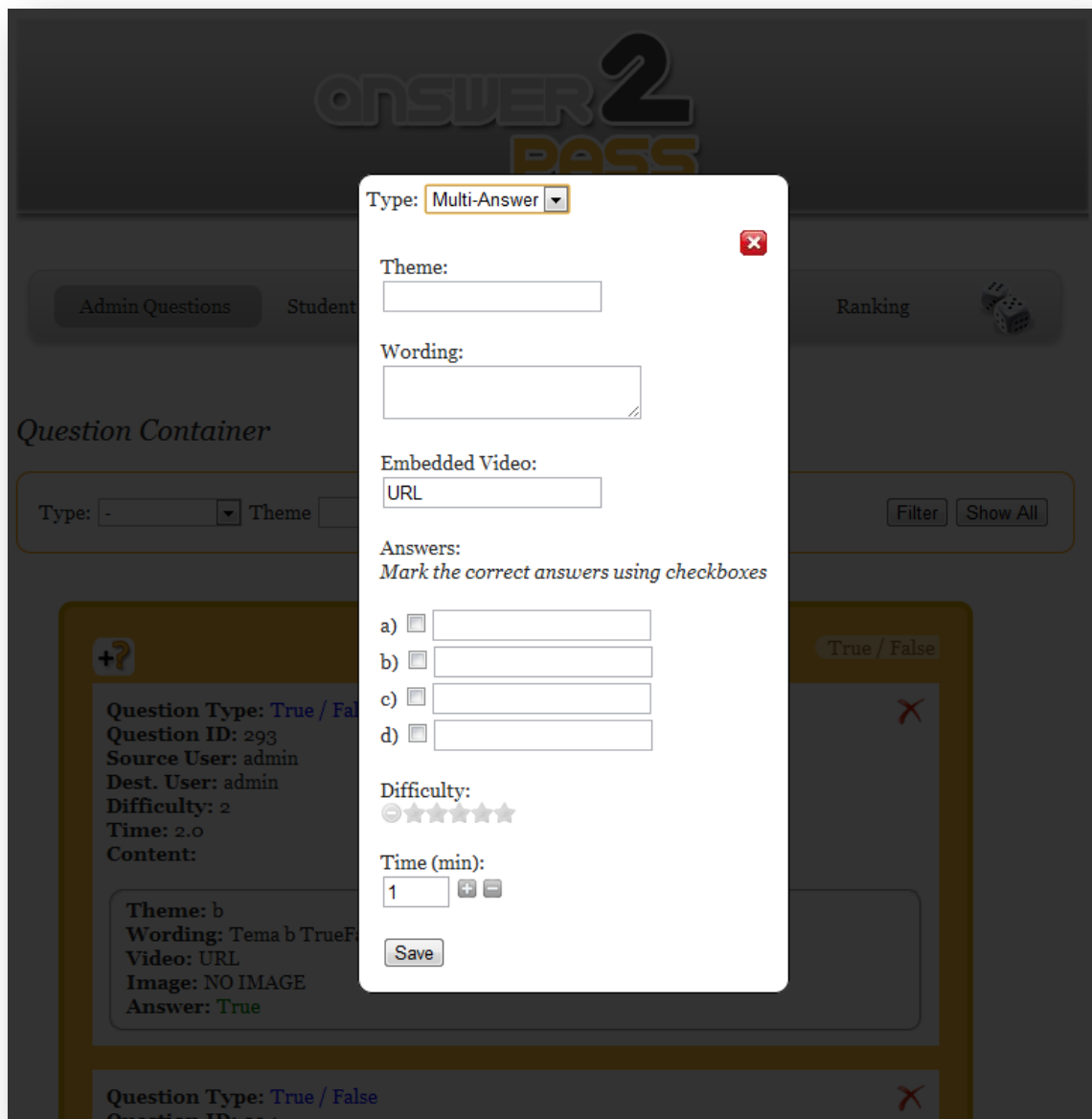


Figura 34: Interfaz de administración VI, Answer2Pass

- **Rankings:** Interfaz destinada a mostrar todos los tipos de rankings al usuario. A través de ella podrá visualizar su posición en la clasificación.

Las acciones que el usuario puede llevar a cabo son:

- **Desplegar rankings,** pulsando sobre el título de cada uno de ellos.

La apariencia de esta interfaz es la siguiente:

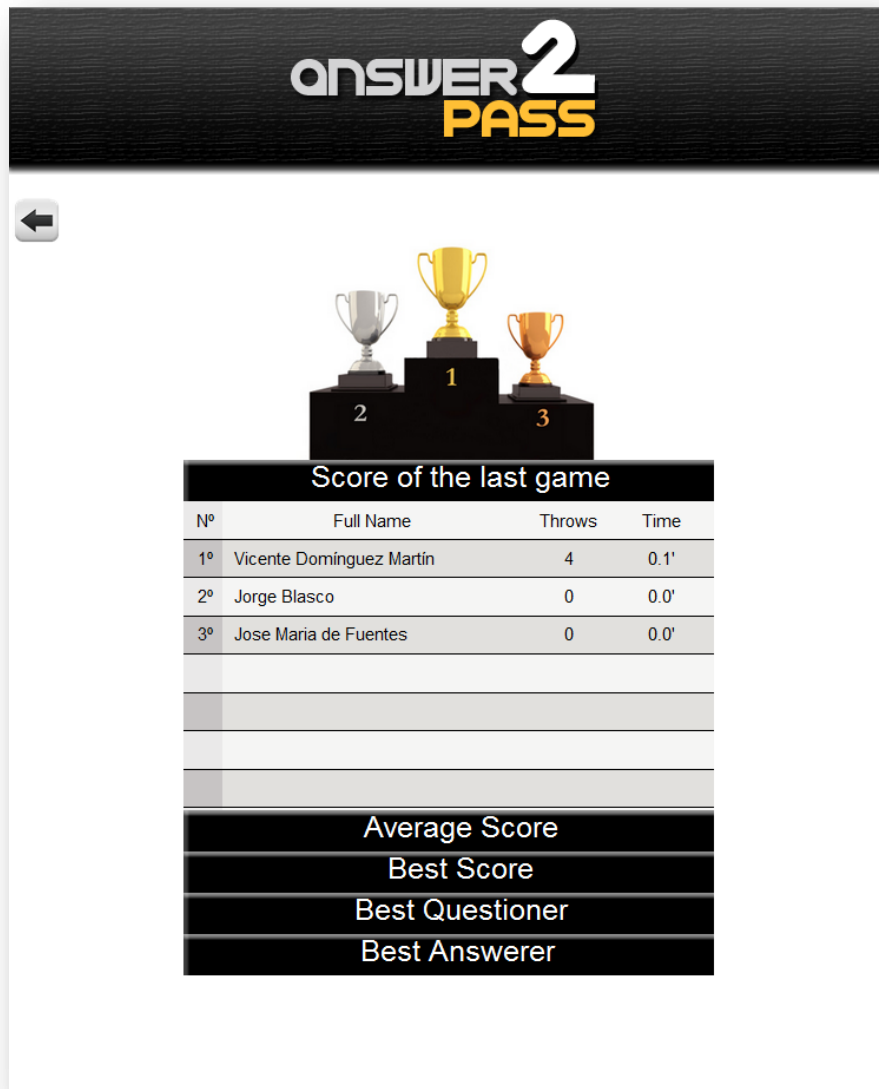


Figura 35: Interfaz de Rankings, Answer2Pass

- **FieldPreview:** Interfaz pensada para que el administrador pueda visualizar todos los tableros que ha creado antes de establecer alguno como tablero de juego. De esta forma puede decidir si le parece adecuado un tipo de tablero o si es más conveniente generar otro nuevo. En la pantalla se muestra el nombre del tablero y la formación del mismo.

Acciones que el administrador puede realizar:

Answer2Pass – Juego de evaluación interactiva de alumnos

- **Volver a la pantalla de administrador**, pulsando el botón denominado “Admin View”.

La apariencia de esta interfaz es la siguiente:



Figura 36: Interfaz de visualización de nuevos tableros, Answer2Pass

Cabe destacar que muchas de las interfaces descritas anteriormente poseen efectos determinados para mostrar sus elementos. Dichos efectos se han implementado sobre los siguientes elementos:

- **Menú de usuario y menú de administrador:** Todos los menús dispuestos en la aplicación muestran un efecto conseguido gracias a la librería *lavalamp.js*¹⁶. Dicho efecto se basa en una región coloreada que se va desplazando acorde al movimiento del ratón por las opciones del menú.
- **Fin de juego:** Al finalizar el juego, un conjunto de efectos simulando fuegos artificiales serán activados durante unos segundos. Esto es conseguido gracias a la librería *fireworks.js*¹⁷.
- **Ventanas emergentes:** Todas las ventanas emergentes como la pantalla de fin de juego, la ventana de preguntas o el creador de las mismas, serán mostradas en el centro, con el fondo blanco y oscureciendo la pantalla principal (dejando esta en un segundo plano). Este efecto es muy parecido al mostrado por Facebook en su visor de fotografías. Cabe destacar que dicho efecto es de programación propia, sin hacer uso de ninguna librería externa.
- **Rankings de juego:** Todos los rankings de juego se podrán visualizar mediante un efecto de tipo acordeón, desplegando uno mientras los otros se ocultan. Esto es conseguido gracias a funciones de la librería *jquery-1.7.1.min.js*¹⁸ (Empleada también para simplificar el desarrollo de funciones en lenguaje JavaScript).
- **Dificultad de las preguntas:** La representación de la dificultad de las preguntas por medio de estrellas se llevará a cabo mediante la librería *jquery.ui-stars.js*¹⁹.
- **Cuenta atrás numérica:** La cuenta atrás numérica dispuesta en todas las ventanas de respuesta de preguntas, ha conseguido su efecto gracias al uso de la librería *jquery.epiclock.js*²⁰.
- **Cuenta atrás gráfica:** El efecto conseguido por la barra de tiempo establecida al responder cualquier pregunta del juego ha sido obtenido de la librería *jquery.progressbar.js*²¹.

¹⁶ Para más información consultar [7]

¹⁷ Para más información consultar [24]

¹⁸ Para más información consultar [12]

¹⁹ Para más información consultar [20]

²⁰ Para más información consultar [6]

²¹ Para más información consultar [25]

3.2.2. Diseño del controlador

A continuación se explicará detalladamente la funcionalidad de todas las clases pertenecientes a la parte del controlador, así como la función que desempeña cada método y atributo incluido en cada una de ellas. Para representar dicha información se van a mostrar una serie de tablas que contendrán los siguientes datos:

- **Clase:** Nombre de la clase del controlador que se va a analizar.
- **Descripción de la clase:** Descripción detallada de la función que desempeña dicha clase en la aplicación.
- **Atributos:** Nombre de todos los atributos de la clase.
- **Descripción de los atributos:** Descripción de la funcionalidad de dichos atributos dentro de la clase.
- **Métodos:** Nombre de todos los métodos implementados dentro de la clase.
- **Descripción de los métodos:** Explicación detallada de todas las acciones que realiza el método, de los parámetros recibidos y de los devueltos (Si devuelve alguno).

La estructura que deben seguir todas las tablas es la siguiente:

Nombre de la clase	
Descripción de la clase	
Atributos	
Nombre del atributo	Descripción del atributo
Métodos	
Nombre del método	Descripción del método

Tabla 5: Formato de la tabla de clases

Las clases implementadas han sido:

SendRequestServlet	
Servlet encargado de enviar peticiones al administrador por parte de los usuarios. Se encarga de almacenar todos los datos recibidos en un objeto de tipo "Petitions".	
Atributos	
uid	Número de identificación del usuario en Facebook
name	Nombre completo del usuario
nia	NIA del usuario
group	Grupo al que pertenece el usuario
Métodos	
doPost(HttpServletRequest, HttpServletResponse)	Método que recibe como parámetro la petición (HttpServletRequest) y la respuesta (HttpServletResponse). Se encarga de recoger todos los datos enviados por el usuario a través del método Post. Es de tipo público y no devuelve ningún valor (Void)

Tabla 6: Clase SendRequestServlet

FieldServlet	
Servlet encargado de gestionar el almacenamiento de la posición de la ficha en el tablero y de las fichas de <i>bonus</i> que el usuario tiene en la partida. Dependiendo de las acciones que el usuario realice, la clase se ocupará de actualizar la nueva posición de la ficha o de añadir o quitar fichas de <i>bonus</i> . Si el juego ha terminado y este <i>servlet</i> es llamado para establecer la nueva posición de la ficha, también se encargará de guardar las puntuaciones correspondientes a la última partida, puntuación media y mejor puntuación (En el caso de que se haya conseguido).	
Atributos	
x	Coordenada X de la nueva posición a establecer en el tablero. Viene dada según la fila de la tabla (en la que se disponen las casillas del tablero) en la que se encuentra la casilla.
y	Coordenada Y de la nueva posición a establecer en el tablero. Viene dada según la columna de la tabla (en la que se disponen las casillas del tablero) en la que se encuentra la casilla.

FieldServlet	
action	String que determina si el usuario ha movido ficha, ha interactuado con alguna ficha de <i>bonus</i> , ha finalizado una partida o ha comenzado una nueva.
action2	String que determina si se debe sumar o restar una ficha <i>bonus</i> al usuario
userid	Número de identificación del usuario en Facebook
endScore	String que contiene la puntuación final de tiradas obtenida
endTime	String que contiene la puntuación final de tiempo obtenida
Métodos	
doPost(HttpServletRequest, HttpServletResponse)	Método que recibe como parámetro la petición (<i>HttpServletRequest</i>) y la respuesta (<i>HttpServletResponse</i>). Se encarga de recoger todos los datos enviados por el usuario a través del método Post. Es de tipo público y no devuelve ningún valor (<i>Void</i>)
timeToDouble(int)	Método que recibe un entero correspondiente a los segundos totales de respuesta y devuelve un <i>double</i> con dichos segundos representados en minutos
timeToSeconds(double)	Método que recibe un <i>double</i> correspondiente a un determinado número de minutos y devuelve un entero con dichos minutos transformados a segundos

Tabla 7: Clase FieldServlet

LoadQuestionServlet	
Servlet encargado de cargar las preguntas cada vez que el usuario hace <i>click</i> en una casilla o en una pregunta sin responder enviada por otro usuario. Si se trata de una casilla de tipo comodín también se encargará de hacer cumplir su función.	
Atributos	
theme	Tema de la pregunta que se debe mostrar
qid	Identificador de la pregunta a mostrar
type	Tipo de la pregunta que se debe mostrar
theme	Tema de la pregunta a mostrar
Métodos	
doPost(HttpServletRequest, HttpServletResponse)	Método que recibe como parámetro la petición (HttpServletRequest) y la respuesta (HttpServletResponse). Se encarga de recoger todos los datos enviados por el usuario a través del método Get. Es de tipo público y no devuelve ningún valor (<i>Void</i>)
random(int)	Método de tipo público que recibe un entero y que devuelve un número aleatorio entre 0 y el parámetro recibido.

Tabla 8: Clase LoadQuestionServlet

ScoreboardServlet	
Servlet encargado de guardar en todo momento las puntuaciones obtenidas en cada tirada. Estas se irán acumulando en atributos del usuario que quedarán registrados en el ranking al finalizar la partida.	
Atributos	
uid	Número de identificación del usuario en Facebook
minutes	Acumulable de minutos que el usuario ha tardado en contestar una pregunta de la partida.
seconds	Acumulable de segundos que el usuario ha tardado en contestar una pregunta de la partida.
throws	Número de tiradas que el usuario lleva en la partida.
Métodos	

ScoreboardServlet	
doPost(HttpServletRequest, HttpServletResponse)	Método que recibe como parámetro la petición (<i>HttpServletRequest</i>) y la respuesta (<i>HttpServletResponse</i>). Se encarga de recoger todos los datos enviados por el usuario a través del método Post. Es de tipo público y no devuelve ningún valor (<i>Void</i>)

Tabla 9: Clase ScoreboardServlet

CheckQuestionsServlet	
<i>Servlet</i> encargado de gestionar las preguntas enviadas por los alumnos. Esta clase accederá a la pregunta por su identificador y realizará los cambios según la acción llevada a cabo por el administrador. Este podrá, a través de esta clase, eliminar una pregunta, corregirla o enviarla al alumno que corresponda, aumentando así la puntuación de 'Mejor Preguntador' del alumno que la creó.	
Atributos	
qid	Identificador de la pregunta a gestionar
typeof	Tipo de la pregunta a gestionar
action	String que determina si la pregunta debe enviarse, borrarse o establecerse como corregida.
questioner	Número identificador del alumno que ha enviado la pregunta
difficulty	Número que representa la dificultad de la pregunta
Métodos	
doPost(HttpServletRequest, HttpServletResponse)	Método que recibe como parámetro la petición (<i>HttpServletRequest</i>) y la respuesta (<i>HttpServletResponse</i>). Se encarga de recoger todos los datos enviados por el usuario a través del método Post. Es de tipo público y no devuelve ningún valor (<i>Void</i>)
increaseQuestionerPoints(String, String)	Método que recibe el identificador del alumno que creó la pregunta y la dificultad de la misma. Se encarga de aumentar la puntuación de dicho alumno en función de la dificultad de la pregunta.

Tabla 10: Clase CheckQuestionsServlet

QuestionServlet	
Servlet encargado del almacenamiento de nuevas preguntas creadas tanto por parte del administrador como de los usuarios. La clase se ocupará de recibir todos los datos especificados por el usuario, y creará un nuevo objeto de pregunta según el tipo.	
Atributos	
typeof	Tipo de la pregunta creada
sourceuser	Identificador de usuario en Facebook que ha creado la pregunta
wording	Enunciado de la pregunta creada
video	Identificador del video de la pregunta creada
answer	Respuesta correcta para preguntas creadas de tipo <i>True/False</i>
time	Tiempo que el usuario ha establecido para la pregunta creada
difficulty	Grado de dificultad que ha elegido un usuario para la pregunta
theme	Tema de la pregunta creada
answers	Array que guarda todas las respuestas correctas seleccionadas por el usuario al crear una pregunta de tipo <i>OneAnswer</i> o <i>MultiAnswer</i> .
possibleanswers	Array que guarda todas las respuestas posibles seleccionadas por el usuario al crear una pregunta de tipo <i>OneAnswer</i> o <i>MultiAnswer</i> .
Métodos	
doPost(HttpServletRequest, HttpServletResponse)	Método que recibe como parámetro la petición (<i>HttpServletRequest</i>) y la respuesta (<i>HttpServletResponse</i>). Se encarga de recoger todos los datos enviados por el usuario a través del método Post. Es de tipo público y no devuelve ningún valor (<i>Void</i>)

Tabla 11: Clase QuestionsServlet

ManageFieldServlet	
Servlet encargado controlar las opciones relacionadas con los tableros creados en la parte del administrador. Esta clase se ocupará de, sí el usuario desea eliminar un tablero, acceder al tablero en cuestión y eliminarlo; y de establecer un número de tablero determinado en todos los usuarios (Si decide seleccionarlo como tablero de juego).	
Atributos	
action	String que determina si un tablero seleccionado debe borrarse o establecerse como tablero de juego
fid	Identificador del tablero
selectedfield	Número del tablero seleccionado en la lista de tableros creados
Métodos	
doPost(HttpServletRequest, HttpServletResponse)	Método que recibe como parámetro la petición (<i>HttpServletRequest</i>) y la respuesta (<i>HttpServletResponse</i>). Se encarga de recoger todos los datos enviados por el usuario a través del método Post. Es de tipo público y no devuelve ningún valor (<i>Void</i>)

Tabla 12: Clase ManageFieldServlet

GenerateFieldServlet	
Servlet encargado de generar tableros. Esta clase recibirá los temas y colores que el usuario desea incluir en el nuevo tablero y se ocupará de generarlo aleatorizando los tipos de casillas (De tema o de comodín) que pueden aparecer en el mismo.	
Atributos	
name	Nombre del tablero a generar
theme1	Tema número uno del tablero a generar especificado por el administrador
theme2	Tema número dos del tablero a generar especificado por el administrador
theme3	Tema número tres del tablero a generar especificado por el administrador
box1	Casilla correspondiente al color del tema 1 marcada o no por el administrador
box2	Casilla correspondiente al color del tema 2 marcada o no por el administrador

GenerateFieldServlet	
box3	Casilla correspondiente al color del tema 3 marcada o no por el administrador
Métodos	
doPost(HttpServletRequest, HttpServletResponse)	Método que recibe como parámetro la petición (<i>HttpServletRequest</i>) y la respuesta (<i>HttpServletResponse</i>). Se encarga de recoger todos los datos enviados por el usuario a través del método Post. Es de tipo público y no devuelve ningún valor (<i>Void</i>)
Random(int)	Método de tipo público que recibe un entero y que devuelve un número aleatorio entre 0 y el parámetro recibido.

Tabla 13: Clase GenerateFieldServlet

ManageRequestServlet	
<i>Servlet</i> encargado de gestionar las peticiones recibidas por parte de los usuarios. Esta clase recibe el identificador de un usuario, se ocupa de identificar si el administrador quiere borrar o aceptar la petición y crea un nuevo objeto o destruye el objeto de petición que tenga como atributo ese identificador de usuario.	
Atributos	
action	String que determina si se debe aceptar o rechazar una petición enviada por un usuario
uid	Número de identificación del usuario en Facebook que envía la petición
name	Nombre del alumno que envía la petición
nia	Número de identificación de alumno del usuario que envía la petición
Métodos	
doPost(HttpServletRequest, HttpServletResponse)	Método que recibe como parámetro la petición (<i>HttpServletRequest</i>) y la respuesta (<i>HttpServletResponse</i>). Se encarga de recoger todos los datos enviados por el usuario a través del método Post. Es de tipo público y no devuelve ningún valor (<i>Void</i>)

Tabla 14: Clase ManageRequestServlet

UserServlet	
Servlet encargado de permitir o denegar el acceso de los usuarios a la aplicación. Posee la funcionalidad de saber el rol del usuario y si su petición ha sido aceptada o denegada.	
Atributos	
uid	Número de identificación del usuario en Facebook que intenta acceder a la aplicación
name	Nombre del usuario que intenta acceder a la aplicación
Métodos	
doPost(HttpServletRequest, HttpServletResponse)	Método que recibe como parámetro la petición (<i>HttpServletRequest</i>) y la respuesta (<i>HttpServletResponse</i>). Se encarga de recoger todos los datos enviados por el usuario a través del método Post. Es de tipo público y no devuelve ningún valor (<i>Void</i>)

Tabla 15: Clase UserServlet

IncreaseAnswerPointsServlet	
Servlet encargado de aumentar la puntuación de ‘Mejor respondedor’ de aquellos alumnos que respondan correctamente las preguntas recibidas por otros.	
Atributos	
uid	Número de identificación del usuario en Facebook que responde la pregunta
qid	Número identificador de la pregunta respondida
typeof	Tipo de la pregunta respondida
Métodos	
doPost(HttpServletRequest, HttpServletResponse)	Método que recibe como parámetro la petición (<i>HttpServletRequest</i>) y la respuesta (<i>HttpServletResponse</i>). Se encarga de recoger todos los datos enviados por el usuario a través del método Post. Es de tipo público y no devuelve ningún valor (<i>Void</i>)

Tabla 16: Clase IncreaseAnswerPointsServlet

CheckAnswersServlet	
Servlet encargado de verificar las respuestas marcadas por los usuarios al responder alguna pregunta del juego. De la misma forma, esta clase será la encargada de calcular la amplitud de la siguiente tirada y de actualizar la puntuación obtenida.	
Atributos	
type	Tipo de la pregunta respondida
qid	Número identificador de la pregunta respondida
selected	Respuestas marcadas por el usuario
action	String que determinará la acción a llevar a cabo si la pregunta respondida procede del tablero de juego o de la pila de preguntas enviadas por los usuarios
uid	Número identificador de Facebook del usuario que envía la pregunta
qTimeM	Minutos de respuesta totales
qTimeS	Segundos de respuesta totales
rTime	Tiempo en el que se ha respondido la pregunta
acTime	Tiempo acumulado en la puntuación
acScore	Tiradas acumuladas en la puntuación
isCorrect	Booleano empleado para indicar si la respuesta es correcta o no
Métodos	
doGet(HttpServletRequest, HttpServletResponse)	Método que recibe como parámetro la petición (<i>HttpServletRequest</i>) y la respuesta (<i>HttpServletResponse</i>). Se encarga de recoger todos los datos enviados por el usuario a través del método Post. Es de tipo público y no devuelve ningún valor (<i>Void</i>)
checkAnswers(String[], String, String[])	Método que recibe las respuestas marcadas por el usuario, la respuesta correcta (<i>True/False</i> o <i>One Answer</i>), y las respuestas correctas (<i>Multi Answer</i>) de la pregunta planteada. Devuelve un booleano, <i>true</i> si la respuesta es correcta y <i>false</i> en el caso contrario.

CheckAnswersServlet	
calculateNextThrow(int, int)	Método que recibe los segundos totales de respuesta y los segundos empleados en responder. Dependiendo de la diferencia existente entre ellos, este método devolverá un entero indicando la amplitud de la siguiente tirada.

Tabla 17: Clase CheckAnswersServlet

3.2.3. Diseño del modelo

A continuación se explicará detalladamente la funcionalidad de todas las clases pertenecientes a la parte del modelo, así como la función que desempeña cada método y atributo incluido en cada una de ellas. Para representar dicha información se van a mostrar una serie de tablas que seguirán el formato establecido en el apartado 3.2.2. Estas son las siguientes:

Petitions	
Clase encargada de almacenar todos los datos de las peticiones enviadas por los usuarios. Implementa métodos que permiten la creación del objeto, el cambio de valores de los atributos y la recuperación de los mismos.	
Atributos	
Key	Clave única asignada al objeto
sourceUser	Identificador de usuario en Facebook que envía la petición
name	Nombre completo del usuario que envía la petición
group	Grupo al que pertenece el usuario que envía la petición
nia	NIA perteneciente al usuario que envía la petición
status	Estado en el que se encuentra el trámite de la petición. Esta podrá estar en curso o aceptada.
Métodos	
Petitions(Params)	Método de tipo constructor que permite la creación del objeto asignando valores a atributos determinados de la clase.

Petitions	
Getters()	Métodos de tipo <i>get</i> que permiten obtener el valor de todos los atributos de la clase
Setters(*)	Métodos de tipo <i>set</i> que permiten modificar el valor de todos los atributos de la clase. Como parámetro recibe el tipo de datos del atributo a cambiar.

Tabla 18: Clase Petitions

User	
Clase encargada de almacenar todos los datos correspondientes a los usuarios de la aplicación. Implementa métodos que permiten la creación del objeto, el cambio de valores de los atributos y la recuperación de los mismos.	
Atributos	
Key	Clave única asignada al objeto
userName	Nombre completo del usuario
userID	Identificador de usuario de Facebook
mail	Correo electrónico del usuario
bestScore	Mejor puntuación en tiradas conseguida por el usuario
bestScoreAux	Auxiliar que va guardando la puntuación de cada partida antes de finalizarla
bestScoreTime	Mejor puntuación en tiempo conseguido por el usuario
bestScoreTimeAux	Auxiliar que va guardando el tiempo de cada partida antes de finalizarla
scoreAverage	Puntuación de tiradas media en las partidas jugadas
timeAverage	Puntuación de tiempo medio en las partidas jugadas
scoreQuestioner	Puntuación del usuario haciendo preguntas
scoreAnswerer	Puntuación del usuario respondiendo preguntas
fieldNumber	Número de tablero de juego asignado
lastposition	Última posición de la ficha en el tablero
questionList	Lista de preguntas que han sido mostradas al usuario
bonusList	Número de fichas bonus que posee el usuario
Métodos	

User	
User(Params)	Método de tipo constructor que permite la creación del objeto asignando valores a atributos determinados de la clase.
Getters()	Métodos de tipo <i>get</i> que permiten obtener el valor de todos los atributos de la clase
Setters(*)	Métodos de tipo <i>set</i> que permiten modificar el valor de todos los atributos de la clase. Como parámetro recibe el tipo de datos del atributo a cambiar.

Tabla 19: Clase User

TrueFalse	
Clase encargada de almacenar todos los datos correspondientes a las preguntas de tipo <i>True/False</i> . Implementa métodos que permiten la creación del objeto, el cambio de valores de los atributos y la recuperación de los mismos.	
Atributos	
Key	Clave única asignada al objeto
sourceUser	Identificador de usuario de Facebook que crea la pregunta
destUser	Identificador de usuario de Facebook a la que va dirigida la pregunta
wording	Enunciado de la pregunta
video	Identificador de video de la pregunta que será utilizado para su reproducción
answer	Respuesta correcta, será <i>true</i> o <i>false</i>
time	Tiempo límite establecido para contestar la pregunta
difficulty	Dificultad de la pregunta
checked	Atributo que identifica si una pregunta ha sido corregida o no
send	Atributo que identifica si una pregunta ha sido enviada o no
theme	Tema del que trata la pregunta
Métodos	
TrueFalse(Params)	Método de tipo constructor que permite la creación del objeto asignando valores a atributos determinados de la clase.

TrueFalse	
Getters()	Métodos de tipo <i>get</i> que permiten obtener el valor de todos los atributos de la clase
Setters(*)	Métodos de tipo <i>set</i> que permiten modificar el valor de todos los atributos de la clase. Como parámetro recibe el tipo de datos del atributo a cambiar.

Tabla 20: Clase TrueFalse

OneAnswer	
Clase encargada de almacenar todos los datos correspondientes a las preguntas de tipo <i>OneAnswer</i> . Implementa métodos que permiten la creación del objeto, el cambio de valores de los atributos y la recuperación de los mismos.	
Atributos	
Key	Clave única asignada al objeto
sourceUser	Identificador de usuario de Facebook que crea la pregunta
destUser	Identificador de usuario de Facebook a la que va dirigida la pregunta
wording	Enunciado de la pregunta
video	Identificador de video de la pregunta que será utilizado para su reproducción
possibleAnswers	Array que almacena todas las respuestas posibles a la pregunta
answer	Respuesta correcta a la pregunta
time	Tiempo límite establecido para contestar la pregunta
difficulty	Dificultad de la pregunta
checked	Atributo que identifica si una pregunta ha sido corregida o no
send	Atributo que identifica si una pregunta ha sido enviada o no
theme	Tema del que trata la pregunta
Métodos	
OneAnswer(Params)	Método de tipo constructor que permite la creación del objeto asignando valores a atributos determinados de la clase.

OneAnswer	
Getters()	Métodos de tipo <i>get</i> que permiten obtener el valor de todos los atributos de la clase
Setters(*)	Métodos de tipo <i>set</i> que permiten modificar el valor de todos los atributos de la clase. Como parámetro recibe el tipo de datos del atributo a cambiar.

Tabla 21: Clase OneAnswer

MultiAnswer	
Clase encargada de almacenar todos los datos correspondientes a las preguntas de tipo <i>MultiAnswer</i> . Implementa métodos que permiten la creación del objeto, el cambio de valores de los atributos y la recuperación de los mismos.	
Atributos	
Key	Clave única asignada al objeto
sourceUser	Identificador de usuario de Facebook que crea la pregunta
destUser	Identificador de usuario de Facebook a la que va dirigida la pregunta
wording	Enunciado de la pregunta
video	Identificador de video de la pregunta que será utilizado para su reproducción
possibleAnswers	Array que almacena todas las respuestas posibles a la pregunta
answers	Array que almacena todas las respuestas correctas a la pregunta
time	Tiempo límite establecido para contestar la pregunta
difficulty	Dificultad de la pregunta
checked	Atributo que identifica si una pregunta ha sido corregida o no
send	Atributo que identifica si una pregunta ha sido enviada o no
theme	Tema del que trata la pregunta
Métodos	
MultiAnswer(Params)	Método de tipo constructor que permite la creación del objeto asignando valores a atributos determinados de la clase.

MultiAnswer	
Getters()	Métodos de tipo <i>get</i> que permiten obtener el valor de todos los atributos de la clase
Setters(*)	Métodos de tipo <i>set</i> que permiten modificar el valor de todos los atributos de la clase. Como parámetro recibe el tipo de datos del atributo a cambiar.

Tabla 22: Clase MultiAnswer

Field	
Clase encargada de almacenar todos los datos relacionados con el tablero de juego. Implementa métodos que permiten la creación del objeto, el cambio de valores de los atributos y la recuperación de los mismos.	
Atributos	
Key	Clave única asignada al objeto
field	Array que almacena la formación del tablero, identificando cada tipo de casilla con un número
name	Nombre del tablero
themes	Array que almacena los temas sobre los que tratan las preguntas del tablero
images	Array que almacena las imágenes de casillas de colores correspondientes a los temas sobre los que tratan las preguntas del tablero
Métodos	
Field(Params)	Método de tipo constructor que permite la creación del objeto asignando valores a atributos determinados de la clase.
Getters()	Métodos de tipo <i>get</i> que permiten obtener el valor de todos los atributos de la clase
Setters(*)	Métodos de tipo <i>set</i> que permiten modificar el valor de todos los atributos de la clase. Como parámetro recibe el tipo de datos del atributo a cambiar.

Tabla 23: Clase Field

3.3. Diagramas de secuencia

En este apartado se van a incluir ejemplos determinados de acciones que pueden llevar a cabo el administrador y el usuario en la aplicación a través de diagramas de secuencia. De esta forma se podrá visualizar claramente las relaciones que existen entre los usuarios y las diferentes partes de la aplicación. Para cada uno de los diagramas de secuencia se incluirá el diagrama en sí y una breve explicación sobre el desarrollo de la acción representada²².

²² Los diagramas que requieran acceder a los datos de un tipo de pregunta determinada tomarán como ejemplo las de tipo True/False.

3.3.1. Envío de una petición al administrador

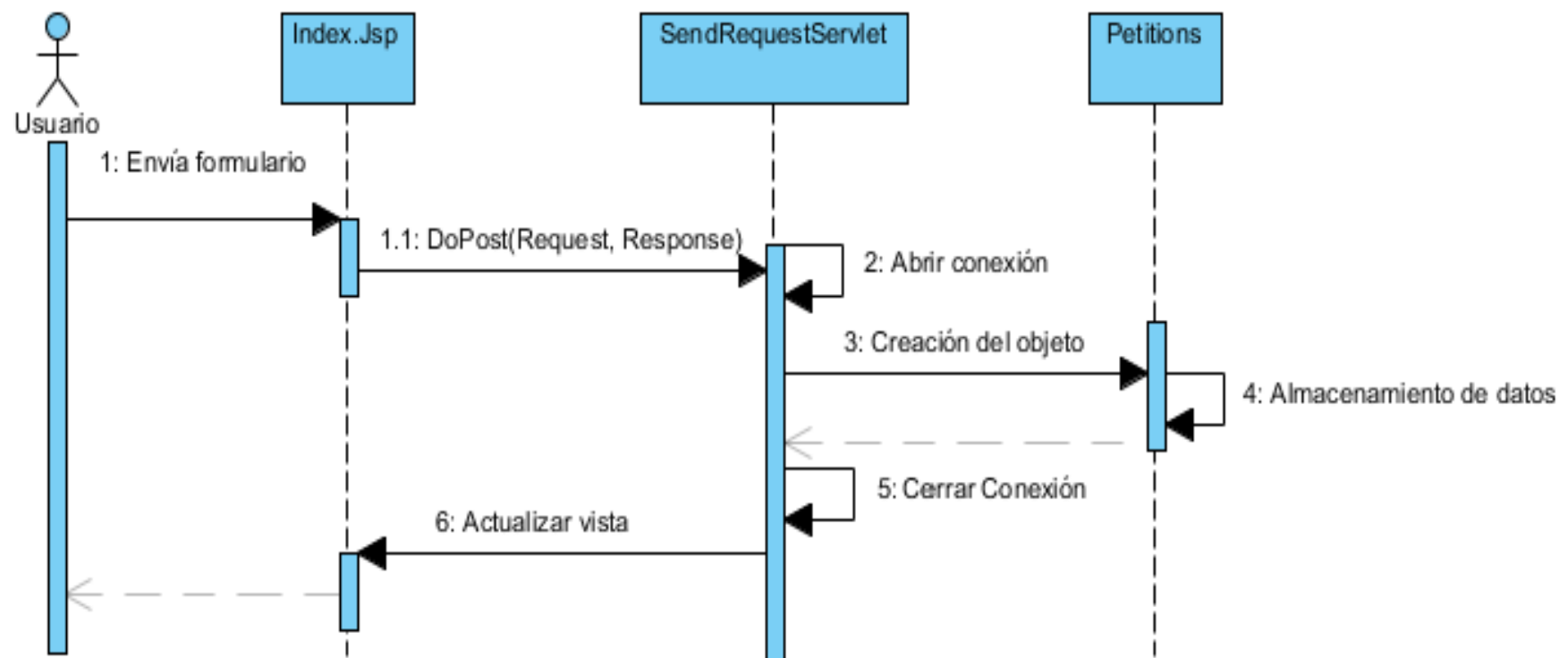


Figura 37: Envío de petición

Con este diagrama se representa la acción de enviar una nueva petición al administrador por parte de un usuario. Sólo aquellos usuarios cuya petición haya sido aceptada por el administrador podrán acceder a la aplicación.

Los pasos realizados para poder llevar a cabo el envío de peticiones son:

- 1- El usuario debe rellenar los campos que se muestran en la interfaz de inicio y pulsar el botón para enviar.
- 2- El *servlet* ocupado de enviar las peticiones recibe todos los parámetros enviados por *post*.
- 3- El *servlet* se ocupa de abrir la conexión para poder almacenar datos, crea el nuevo objeto de tipo *Petitions*, y cierra de nuevo la conexión.
- 4- El *servlet* devuelve la vista al usuario.

3.3.2. Aceptar una petición

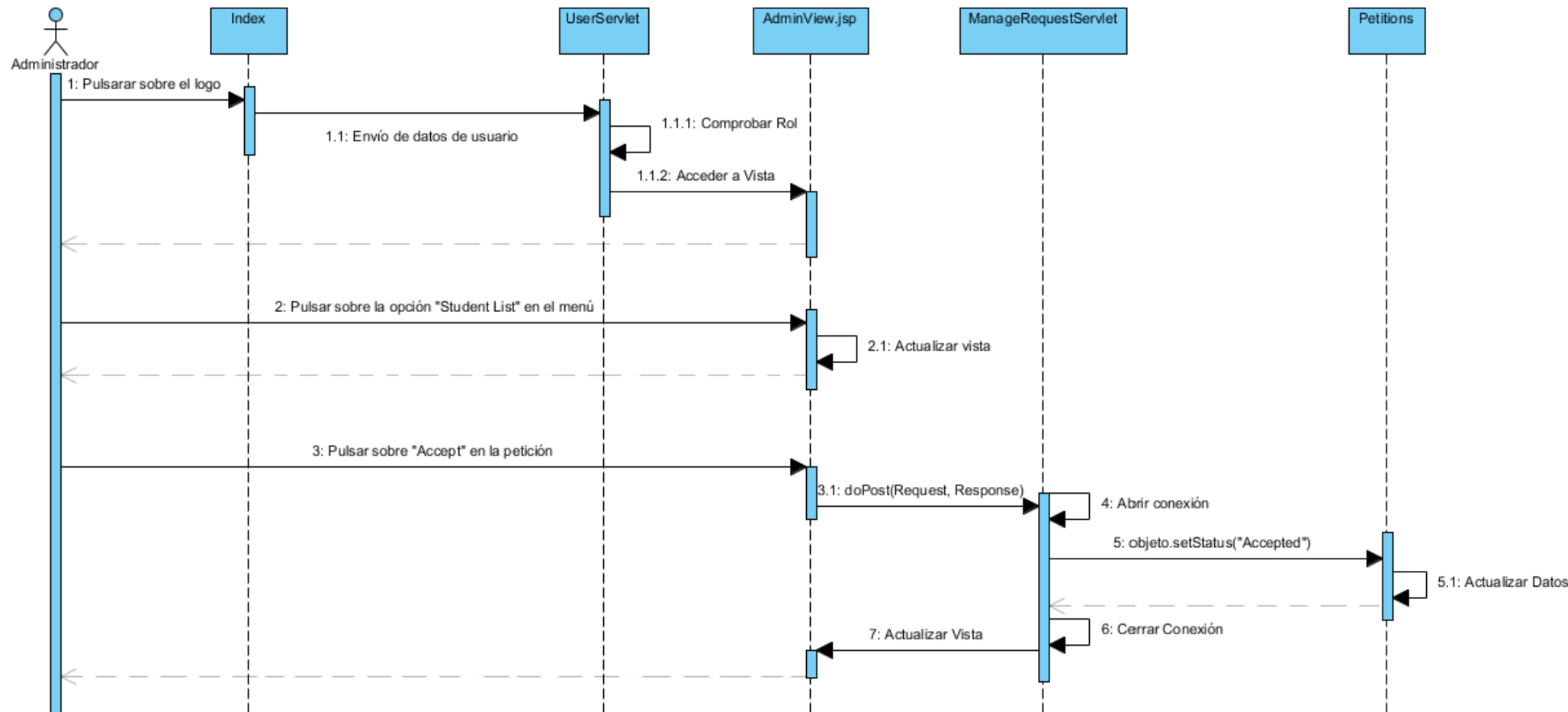


Figura 38: Aceptar petición

Answer2Pass – Juego de evaluación interactiva de alumnos

Con este diagrama se representa la acción que realiza el administrador de aceptar una petición de nuevo usuario. Una vez que la petición se acepte, el usuario pasará a la lista de usuarios registrados y podrá acceder a la aplicación.

Los pasos realizados para poder aceptar peticiones son:

- 1- El administrador pulsa sobre el logo de la aplicación en la interfaz principal para poder acceder.
- 2- El *servlet* ocupado de controlar el acceso comprueba el rol de usuario y redirige a la siguiente pantalla, la vista de administrador.
- 3- Una vez el administrador ha visualizado la interfaz, pulsa en el botón del menú ocupado de mostrar la lista de usuario que han enviado una petición.
- 4- Se muestra la lista.
- 5- El administrador acepta la petición.
- 6- El *servlet* encargado de gestionar las peticiones recibe todos los parámetros enviados por *post*.
- 7- El *servlet* se encarga posteriormente de abrir la conexión, acceder al objeto de tipo *Petitions* y de cambiar su estado a través del método *set*.
- 8- El *servlet* cierra la conexión y actualiza la vista del administrador mostrando los cambios producidos.

3.3.3. Crear una nueva pregunta

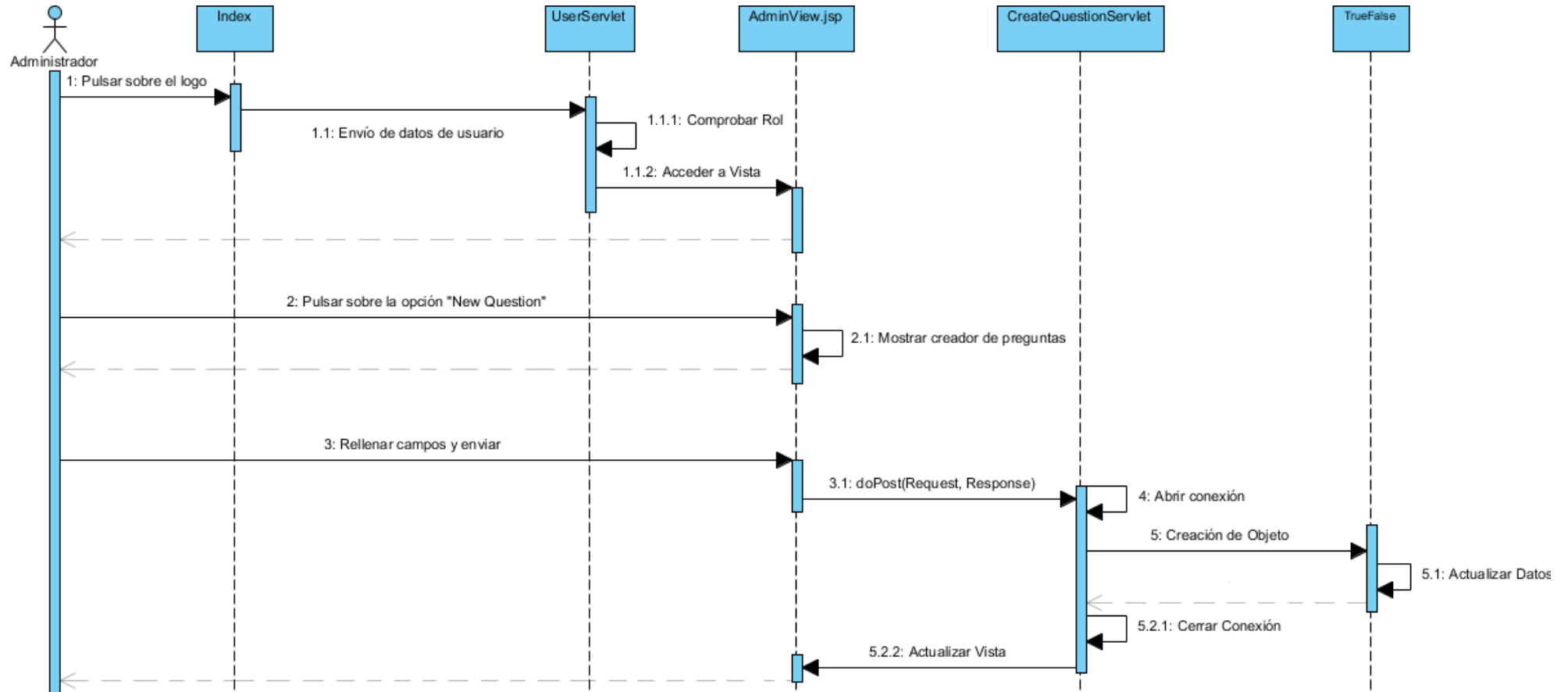


Figura 39: Crear nueva pregunta

Con este diagrama se representa la acción que realiza el administrador de crear una nueva pregunta. Una vez que la acción se lleve a cabo, la pregunta pasará a ser visible en el banco de preguntas creadas. Destacar que se ha tomado como ejemplo la creación de una pregunta de tipo *True/False*.

Los pasos realizados para poder llevar a cabo la creación de preguntas son:

- 1- El administrador pulsa sobre el logo de la aplicación en la interfaz principal para poder acceder.
- 2- El *servlet* ocupado de controlar el acceso comprueba el rol de usuario y redirige a la siguiente pantalla, la vista de administrador.
- 3- Una vez el administrador ha visualizado la interfaz, pulsa en el botón habilitado para crear nuevas preguntas.
- 4- Se muestra la ventana del creador de preguntas.
- 5- El administrador rellena todos los campos y envía el formulario.
- 6- El *servlet* encargado de crear nuevas preguntas recibe todos los parámetros enviados por *post*.
- 7- El *servlet* se encarga posteriormente de abrir la conexión y de crear un nuevo objeto de tipo *True/False*.
- 8- El *servlet* cierra la conexión y actualiza la vista del administrador mostrando los cambios producidos.

3.3.4. Responder una pregunta del tablero

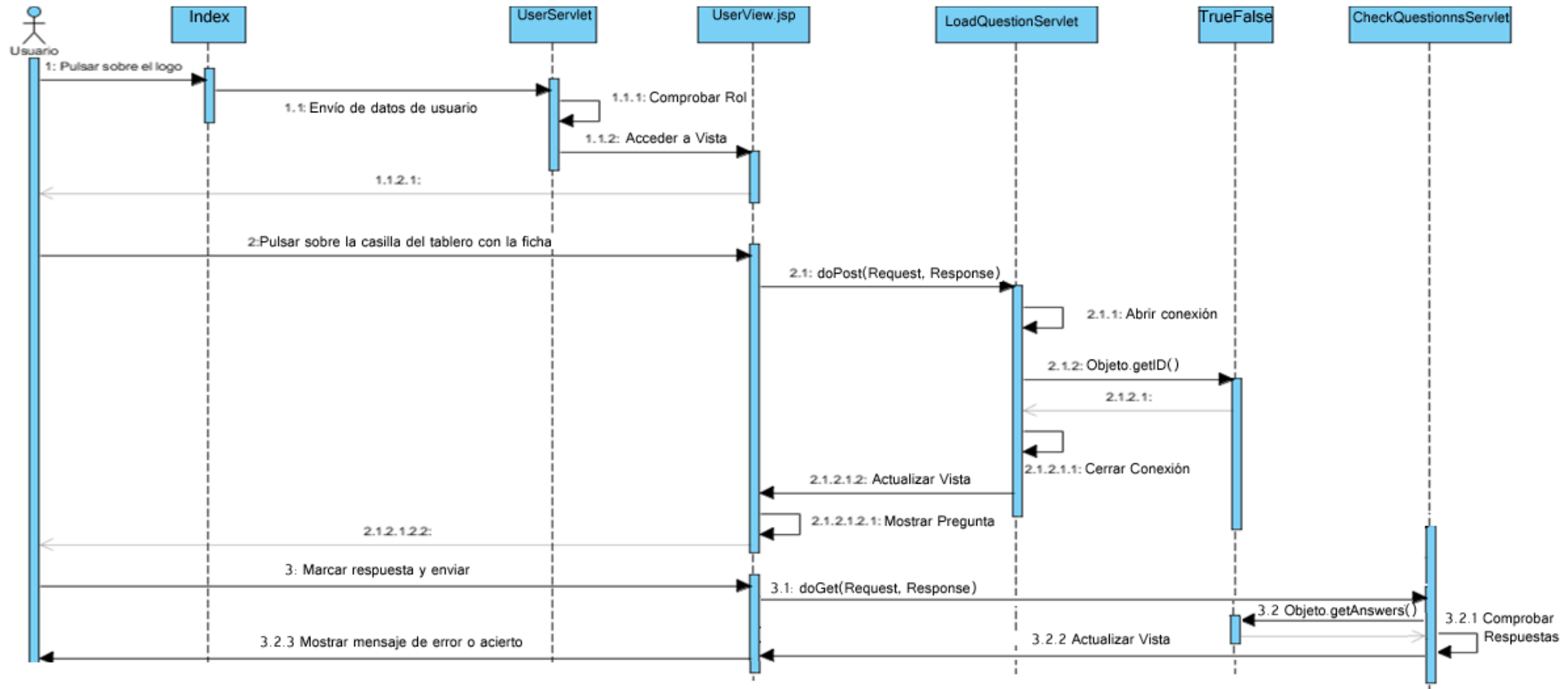


Figura 40: Responder una pregunta

Answer2Pass – Juego de evaluación interactiva de alumnos

Con este diagrama se representa la acción que realiza un usuario de la aplicación al responder una pregunta del tablero. Destacar que se ha tomado como ejemplo la respuesta a una pregunta de tipo *True/False* y que se ha omitido la parte de calcular y guardar la nueva posición en el tablero y la de guardar las puntuaciones conseguidas.

Los pasos realizados para poder responder una pregunta son:

- 1- El usuario pulsa sobre el logo de la aplicación en la interfaz principal para poder acceder.
- 2- El *servlet* ocupado de controlar el acceso comprueba el rol de usuario y redirige a la siguiente pantalla, la vista de usuario.
- 3- Una vez el usuario ha visualizado la interfaz, pulsa sobre la casilla del tablero que contiene la imagen de la ficha de juego
- 4- Se envía la información a un *servlet* encargado de recibir los parámetros y obtener la pregunta.
- 5- El *servlet* abre la conexión y recupera la pregunta tipo *True/False* a mostrar.
- 6- El *servlet* cierra la conexión y actualiza la vista del usuario mostrando la ventana que contiene la pregunta.
- 7- El usuario marca la respuesta que piensa que es la correcta y la envía.
- 8- La clase *CheckAnswersServlet* recibe las respuestas marcadas por el alumno y el identificador de la pregunta.
- 9- Ese mismo *servlet*, se ocupa de recuperar las respuestas correctas a la pregunta de la clase *True/False* y las compara con las enviadas por el alumno.
- 10- Finalmente se actualiza la vista mostrando al usuario si ha acertado o no la pregunta.

4. Implementación

A continuación se explicará detalladamente la implementación del sistema desarrollado. Por un lado se detallarán los aspectos de la implementación considerados más relevantes para el desarrollo de la aplicación, y por otro se enunciarán los resultados obtenidos una vez concluido el plan de pruebas establecido en el punto de 'Análisis'.

4.1. Aspectos de la implementación

En este apartado de la implementación del sistema se van a explicar de forma detallada los aspectos más importantes para el desarrollo. Principalmente se analizará, añadiendo fragmentos de código cuando sea necesario, la conexión de la aplicación con Facebook y la implementación de JDO como sistema de almacenamiento.

4.1.1. Conexión con Facebook

Como se ha mencionado en el capítulo de 'Análisis', Facebook es una red social que permite a sus usuarios programar aplicaciones que puedan estar conectadas a la misma. De esta forma, a través de una serie de funciones de sus librerías, se pueden intercambiar datos que posibilitan la obtención de información de los contactos y la ejecución de funciones propias de la red social como publicar en el muro o el envío de notificaciones.

Antes de que la aplicación pueda interactuar con los datos del usuario y realizar acciones propias de la red social, es necesario que este haya pasado el proceso de autenticación y haya dado los permisos necesarios a la aplicación para acceder a sus datos. Si estos dos pasos no se completan de forma correcta se impedirá al usuario el acceso al juego, ya que no se podrán llevar a cabo varias acciones pensadas para interactuar con la red social.

A continuación se va a mostrar el diagrama del proceso descrito (ver figura 41). En él se especifica el comportamiento del sistema ante el estado de autenticación del usuario dentro de la aplicación. Como se puede observar, el usuario comienza introduciéndose en la aplicación mediante una notificación o el repositorio de aplicaciones de Facebook. Una vez que está dentro y quiere acceder al juego, se comprueba en primer lugar si está identificado en la red social. Si no es así, a través del botón habilitado para la función, se le muestra la ventana de *login* para que introduzca correo electrónico y contraseña. Por el contrario, se pasa a comprobar si el usuario ha dado permisos a la aplicación para acceder a sus datos. De nuevo, si esto no se ha llevado a cabo anteriormente, se mostrará una ventana para poder realizar dicha

acción. Finalmente, si el usuario ha dado permisos y se encuentra identificado en Facebook se le permite el acceso a la aplicación, iniciando así el intercambio de datos.

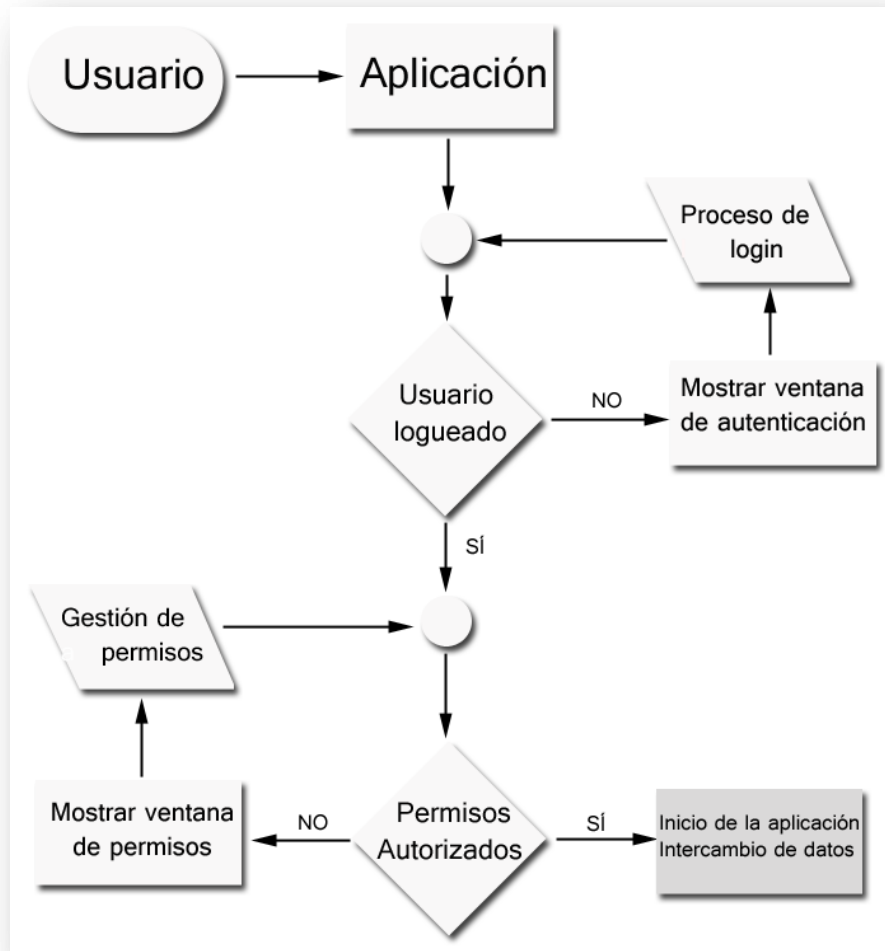


Figura 41: Proceso de acceso a la aplicación

Este proceso, llevado a cabo al comenzar la aplicación, ha sido implementado en primer lugar autenticando la aplicación con el número de identificación proporcionado por Facebook (En el código 'appId') al verificar la cuenta. Una vez se ha producido la verificación, se procede a la extracción de la librería Javascript llamada 'all.js'. El código empleado para realizar dichas acciones se puede observar a continuación:

```
window.fbAsyncInit = function() {  
  FB.init({  
    appId      : '380602298621072', // App ID  
    channelUrl : 'http://answer2pass.appspot.com/', // Channel File  
    status     : true, // check login status  
    cookie     : true, // enable cookies to allow the server to access the session  
    xfbml      : true, // parse XFBML  
  });  
};
```

Número identificativo de la aplicación

Figura 42: Autenticación con Facebook²³

```
// Load the SDK Asynchronously  
(function(d){  
  var js, id = 'facebook-jssdk'; if (d.getElementById(id)) {return;}  
  js = d.createElement('script'); js.id = id; js.async = true;  
  js.src = "http://connect.facebook.net/en_US/all.js";  
  d.getElementsByTagName('head')[0].appendChild(js);  
})(document);
```

Ruta de la librería

Figura 43: Obtención de librería Javascript²⁴

Después de obtener la librería Javascript, se comprobará si este ha pasado el proceso de autenticación de Facebook y si ha dado permisos a la aplicación para acceder a sus datos. Esto se ha implementado de la siguiente forma:

²³ Función obtenida de las propuestas en el apartado de desarrolladores de Facebook [5]

²⁴ Función obtenida de las propuestas en el apartado de desarrolladores de Facebook [5]

```
FB.getLoginStatus(function(response) {  
  
  if (response.status === 'connected') {  
    // the user is logged in and connected to your  
    // app, and response.authResponse supplies  
    // the user's ID, a valid access token, a signed  
    // request, and the time the access token  
    // and signed request each expire  
    //var uid = response.authResponse.userID;  
    uid = response.authResponse.userID;  
    var accessToken = response.authResponse.accessToken;  
    perm = true;  
  
  } else if (response.status === 'not_authorized') {  
    // the user is logged in to Facebook,  
    //but not connected to the app  
    alert("Hi user , you are logged but not authorized on this app");  
  
  } else {  
    // the user isn't even logged in to Facebook.  
    alert("Hi user , you aren't logged in");  
  }  
});
```

Obtención del identificador de usuario y del token de acceso

Figura 44: Comprobar estado de autenticación²⁵

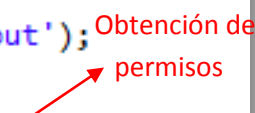
Como se puede observar en el fragmento de código anterior, se comprueba la propiedad 'status' de la respuesta recibida al realizar la petición a Facebook. De esta forma, si el estado es 'connected' se obtendrá el *token* de acceso del usuario y su identificador (Usado posteriormente para obtener cualquier dato o para identificar a quién enviar una petición). Por el contrario, si el estado recibido es 'not_authorized' o cualquier otro, se alertará al usuario del estado de autenticación que tiene en la aplicación.

Si el usuario no es capaz de acceder a la aplicación porque su estado es diferente de 'connected', tendrá que pulsar sobre un botón habilitado para mostrar la ventana de *login* y la de permisos (Si no está identificado en Facebook) o sólo la de permisos (Si está identificado en la red social). La funcionalidad de este botón está implementada de la siguiente forma:

²⁵ Función obtenida de las propuestas en el apartado de desarrolladores de Facebook [5]

```
//Permissions
function getPerms() {

    var cb = function(response) {
        Log.info('FB.login callback', response);
        if (response.status === 'connected') {
            Log.info('User logged in');
        } else {
            Log.info('User is logged out');
        }
    };
    FB.login(cb, { scope: 'publish_stream' });
};
```

Figura 45: Obtención de permisos²⁶

Como se observa en la imagen anterior, el permiso necesario para el intercambio de datos pensado en la aplicación es el de 'publish_stream', que según un apartado dispuesto en la página de desarrolladores de Facebook²⁷ es el que permite realizar acciones como postear en el muro.

Una vez que el usuario pueda acceder al juego, tendrá la posibilidad de interactuar con la red social llevando a cabo acciones como publicar en el muro la puntuación obtenida en la última partida o, en el caso de ser administrador, enviando notificaciones a los usuarios cada vez que una pregunta dirigida hacia ellos sea corregida y enviada o notificando alguna incidencia o comentario sobre las peticiones de usuario recibidas. La implementación de estas funcionalidades se ha llevado a cabo de la siguiente forma:

²⁶ Función obtenida de las propuestas en el apartado de desarrolladores de Facebook [5]

²⁷ Para más información consultar [5]


```
//Send notification
function notify(userID, message){

    var message;
    if(message == 1)
        message='Someone has sent you a question';
    else
        message=$("#"+userID+"_message").attr("value");

    window.scrollTo(0,0);

    FB.ui({method: 'apprequests',
           message: message,
           to: userID
    });

    return false;
}
```

Figura 46: Envío de notificaciones²⁸

Como se puede observar, el código de la imagen anterior corresponde con el de la función de envío de notificaciones a los usuarios por parte del administrador. Como se ha mencionado anteriormente, este podrá notificar comentarios e incidencias de las peticiones recibidas o notificar el envío de preguntas realizadas por otros usuarios. Esta función recibe dos parámetros, uno el identificador de usuario y otro el mensaje que se desea enviar. Si el parámetro correspondiente al mensaje es igual a 1, significa que el administrador está enviando una notificación de envío de pregunta, por lo que se formará el mensaje *'Someone has sent you a question'* y se enviará al usuario que posea el identificador recibido como primer parámetro. Por el contrario, si el parámetro correspondiente al mensaje es un número distinto de 1, significa que el administrador quiere notificar un comentario o incidencia de las peticiones recibidas, por lo que se obtendrá el valor del mensaje (Botón de tipo entrada de texto) con JQuery (JavaScript) y se enviará al usuario especificado en el primer parámetro.

²⁸ Función editada a partir de las propuestas en el apartado de desarrolladores de Facebook [5]

```
//Post game score in the wall
function postScore (score){

    FB.publish({
        message : "The score obtained in Answer2Pass was "+score,
        name : "Answer2Pass score",
        description : "Score type: Last game"
    },function(published){
        if (published)
            alert("Published correctly!");
        else
            alert("Error:: Unpublished!");
    });
}
```

Figura 47: Publicar puntuación en el muro²⁹

La función de la imagen anterior se corresponde con la de postear puntuaciones en el muro. A través de dicha función el usuario puede compartir con la red social la puntuación que ha obtenido al finalizar una partida de Answer2Pass. Únicamente recibe un parámetro, una cadena formada con la puntuación obtenida. Esta será enviada junto con un mensaje predefinido, un nombre y una descripción. Una vez realizado el posteo, se notificará al usuario mediante un mensaje de alerta si se ha realizado de forma correcta o ha sucedido algún error imprevisto.

4.1.2. JDO

A continuación se va a explicar detalladamente la implementación del sistema de almacenamiento de datos empleado en el desarrollo de la aplicación. Para ello, al igual que en el apartado anterior, se van a ir mostrando imágenes correspondientes con fragmentos de código explicando su funcionalidad.

El primer paso sería definir las clases de datos con JDO. En ellas se encontrarán todas las definiciones de atributos, especificación de clases, constructores que den valor a los mismos al crear el objeto y por último, los métodos que permitirán dar funcionalidad a la clase, como es el ejemplo de los 'GET', para recuperar el valor de los atributos, y el de los 'SET', para cambiar el valor.

²⁹ Función editada a partir de las propuestas en el apartado de desarrolladores de Facebook [5]

```
import com.google.appengine.api.datastore.Key;
import com.google.appengine.api.users.User;

import javax.jdo.annotations.IdGeneratorStrategy;
import javax.jdo.annotations.PersistenceCapable;
import javax.jdo.annotations.Persistent;
import javax.jdo.annotations.PrimaryKey;

@SuppressWarnings("unused")
@PersistenceCapable
public class Users {
    @PrimaryKey
    @Persistent(valueStrategy = IdGeneratorStrategy.IDENTITY)
    private Key key;

    @Persistent
    private String userName;

    @Persistent
    private String userID;

    @Persistent
    private String role;

    @Persistent
    private String mail;
}
```

Figura 48: Definición de clases JDO I

En la imagen anterior podemos observar la primera parte del cuerpo de la clase de datos. En el código representado vemos en primer lugar las librerías importadas, que en este caso son las de Google App-Engine y las de JDO de tipo 'javax.jdo*'. Estas últimas se utilizan para indicar anotaciones de JDO en la clase mediante el carácter '@'. De esta forma, cuando una clase se quiere poder almacenar y recuperar de la base de datos tiene que estar precedida de la anotación `@PersistenceCapable`, al igual que los atributos que quieran almacenarse tienen que poseer la anotación `@Persistent` (Anotación `@NotPersistent` para aquellos que no desean ser almacenados). Por otro lado, los atributos se definen siguiendo la estructura: Especificador de control de acceso – Tipo de datos – Nombre del atributo, donde el primero siempre será privado al solo querer acceder a los atributos dentro de la misma clase. Si cualquiera de estos atributos quiere convertirse en clave primaria, hay que hacer uso de la notificación `@PrimaryKey` seguida de `@Persistent(valueStrategy = IdGeneratorStrategy.IDENTITY`. De

esta manera todos los atributos que las posean generarán automáticamente un identificador único que irá aumentando su valor en 1 cada vez que se instancie un objeto de la clase. Cabe destacar que, en la clase destinada al almacenamiento de datos de usuario, aunque esté claro que el identificador de Facebook ya es único, se ha decidido añadir una nueva clave primaria por precaución ante cualquier cambio producido en la red social, ya que es un dato ajeno a la aplicación y puede cambiar en cualquier momento.

El diagrama muestra un fragmento de código Java para la definición de la clase `Users`. Las anotaciones en rojo indican:
- **Declaración del constructor**: apunta a la línea `public Users`.
- **Parámetros recibidos**: apunta a la lista de parámetros entre paréntesis.
- **Asignación de valores**: apunta a las líneas de asignación de valores a los atributos `this`.

```
public Users(String userID, String userName, String mail, String role,
    int bestScore, double bestScoreTime, int gameScore,
    double gameScoreTime, int gameScoreAux, double gameScoreTimeAux,
    int scoreAverage, double timeAverage, int scoreQuestioner,
    int scoreAnswerer, int fieldNumber, int[] lastPosition,
    int bonusList) {

    this.userName = userName;
    this.userID = userID;
    this.mail = mail;
    this.role = role;

    // Ranking
    this.bestScore = bestScore;
    this.bestScoreTime = bestScoreTime;
    this.gameScore = gameScore;
    this.gameScoreTime = gameScoreTime;
    this.gameScoreAux = gameScoreAux;
    this.gameScoreTimeAux = gameScoreTimeAux;
    this.scoreAverage = scoreAverage;
    this.timeAverage = timeAverage;
    this.scoreQuestioner = scoreQuestioner;
    this.scoreAnswerer = scoreAnswerer;

    // Game config
    this.fieldNumber = fieldNumber;
    this.lastPosition = lastPosition;
    this.bonusList = bonusList;
}
```

Figura 49: Definición de clases JDO II

En el fragmento de código anterior se puede observar el constructor de la clase destinada al almacenamiento de datos de los usuarios. A través de él se reciben todos los parámetros enviados al crear el objeto de la clase y se guardan en los atributos de la misma. Este método es el responsable del almacenamiento de todos los datos iniciales del usuario.

Todos los datos almacenados gracias al método mencionado anteriormente deben poder recuperarse y modificarse de la base de datos. Para ello se deben implementar los métodos 'GET' y 'SET' de cada uno de los atributos. La imagen que muestra un ejemplo de fragmento de código correspondiente a esta parte es la siguiente:

```
public int getFieldName() {  
    return fieldName;  
}  
  
public void setFieldName(int fieldName) {  
    this.fieldName = fieldName;  
}  
  
public int[] getLastPosition() {  
    return lastPosition;  
}  
  
public void setLastPosition(int lastPosition[]) {  
    this.lastPosition = lastPosition;  
}
```

Figura 50: Definición de clases JDO III

Como se puede observar, el método 'GET' no recibe ningún parámetro, ya que lo único necesario es obtener el atributo. Por el contrario, el método 'SET' recibe el valor por el que se quiere sustituir el último guardado en el atributo, por lo que al igual que en el constructor asigna el nuevo valor.

Una vez declarada la clase de datos, para poder almacenar un objeto es necesario instanciarlo previamente. Esto se lleva a cabo mediante el término 'new' y la especificación de parámetros que se desean enviar al constructor de la clase. La imagen siguiente, muestra un fragmento de código responsable de recibir peticiones por parte de los usuarios y de almacenarlas en la base de datos. Como se puede observar, en primer lugar se reciben todos los datos enviados por el usuario. Después se obtiene el '*PersistenceManager*', que será el encargado del almacenamiento de datos mediante la llamada a la función '*makePersistent*'. Posteriormente se crea el nuevo objeto de la clase peticiones insertando los valores recibidos por el usuario como parámetro y, finalmente, se realiza el almacenamiento y el cierre del objeto '*PersistenceManager*'. Cabe destacar que este último paso se lleva a cabo dentro de un '*try – finally*' por si se produce alguna excepción.

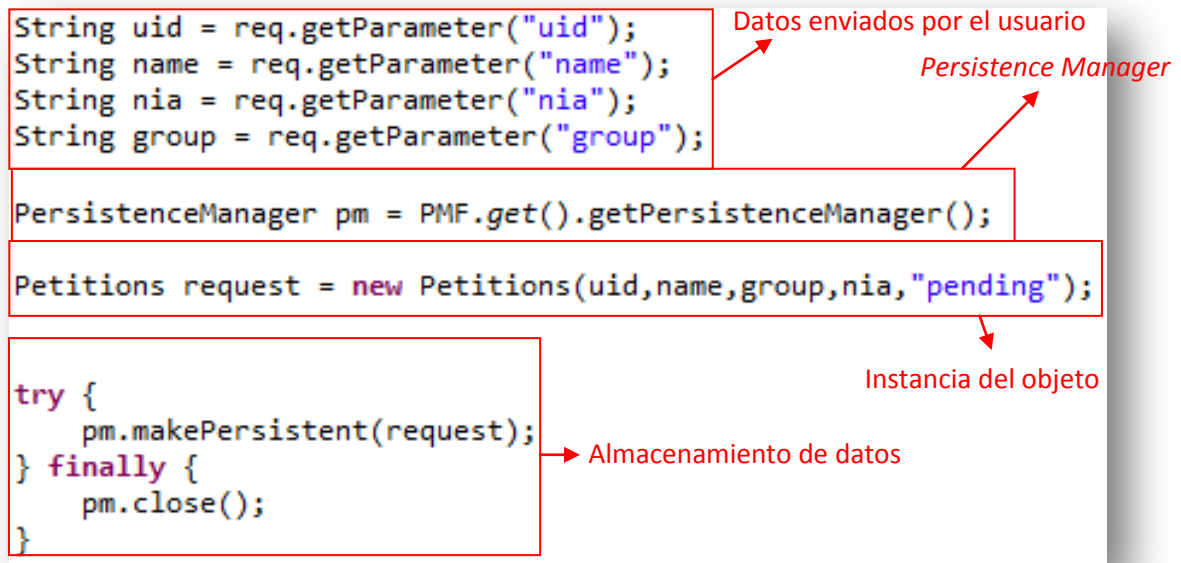


Figura 51: JDO, creación de objetos

Una vez se tienen todos los datos almacenados se puede acceder a ellos y modificarlos si la situación lo requiere. Todas estas consultas se llevan a cabo en muy pocas líneas de código con JDO, evitando así costosas consultas en otros lenguajes como SQL. Se puede acceder a los datos de varias formas, entre ellas está el acceso directo a un objeto concreto mediante su clave primaria o la recuperación de todos los objetos almacenados de una misma clase. Una vez que un objeto se recupera se puede acceder a los atributos del mismo haciendo uso de los métodos 'Get' y 'SET' mencionados anteriormente. Las siguientes imágenes muestran fragmentos de código correspondientes a este tipo de consultas y acceso a datos:

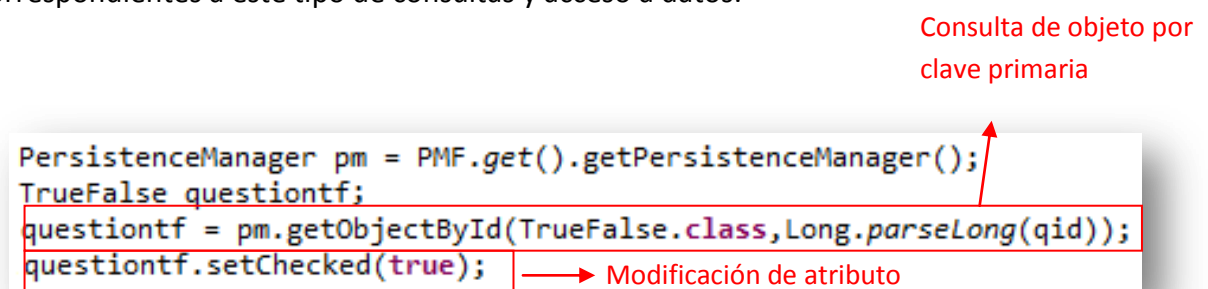


Figura 52: JDO, Consulta por clave primaria

En esta primera imagen se observa la modificación de datos con JDO. En primer lugar se obtiene el objeto correspondiente a la clave primaria enviada como parámetro de la función *'getObjectByld'* del *'Persistence Manager'*. Finalmente basta con añadir una línea donde se llame al método *'SET'* implementado para el atributo que desea modificar.

```
PersistenceManager pm = PMF.get().getPersistenceManager();
String query = "select from " + Petitions.class.getName();
List<Petitions> request = (List<Petitions>) pm.newQuery(query).execute();
for (Petitions g : request) {
    pm.deletePersistent(g);
}
```

Eliminar todos los elementos de la lista de objetos de petición

Consulta de todos los objetos de la clase de peticiones

Figura 53: JDO, consulta y eliminación de un grupo de objetos

En esta otra imagen se forma en primer lugar una cadena que contiene la consulta a todos los objetos de tipo petición. Después se obtiene mediante esa consulta una lista formada por elementos de tipo objeto petición. Finalmente con un bucle de tipo *'for'* se recorre dicha lista y se tratan los datos de cada objeto (En este caso se eliminan de la base de datos).

4.2. Resultados de las pruebas

Una vez finalizado el desarrollo de la aplicación y pasado la batería de pruebas se observa que los resultados han sido satisfactorios, la ejecución de todas las pruebas se ha llevado a cabo de forma correcta. La tabla que representa los resultados obtenidos en dichas pruebas es la siguiente:

Prueba	Resultado	Prueba	Resultado
PA - 01	Correcto	PA - 21	Correcto
PA - 02	Correcto	PA - 22	Correcto
PA - 03	Correcto	PA - 23	Correcto
PA - 04	Correcto	PA - 24	Correcto
PA - 05	Correcto	PA - 25	Correcto
PA - 06	Correcto	PA - 26	Correcto
PA - 07	Correcto	PA - 27	Correcto
PA - 08	Correcto	PA - 28	Correcto
PA - 09	Correcto	PA - 29	Correcto
PA - 10	Correcto	PA - 30	Correcto
PA - 11	Correcto	PA - 31	Correcto
PA - 12	Correcto	PA - 32	Correcto
PA - 13	Correcto	PA - 33	Correcto
PA - 14	Correcto	PA - 34	Correcto
PA - 15	Correcto	PA - 35	Correcto
PA - 16	Correcto	PA - 36	Correcto
PA - 17	Correcto	PA - 37	Correcto
PA - 18	Correcto	PA - 38	Correcto
PA - 19	Correcto	PA - 39	Correcto
PA - 20	Correcto	PA - 40	Correcto

Tabla 24: Resultados de las pruebas

5. Gestión del proyecto

En este apartado se pretende explicar detalladamente todos los aspectos que forman la gestión del proyecto. Para ello se van a distinguir tres puntos claramente diferenciados: Primero, se van a explicar cada una de las fases planificadas por las que ha pasado el proyecto, y se realizará una comparación analizando la planificación inicial y la que realmente se ha llevado a cabo; en segundo lugar, se van a especificar todas las herramientas *software* y *hardware* que han sido necesarias para el desarrollo de la aplicación; y por último, se va a realizar un análisis exhaustivo donde se verá reflejado el coste que ha supuesto el desarrollo de la aplicación (haciendo uso de los precios propuestos en la plantilla de presupuestos proporcionada) y los posibles beneficios obtenidos si esta se introdujera en el mercado.

5.1. Planificación del proyecto

En este primer apartado se detallarán todas las tareas realizadas a lo largo del proyecto y se realizará una comparación entre la planificación inicial y final del mismo tomando como referencia el diagrama de Gantt elaborado en la primera fase.

5.1.1. Planificación inicial

En primer lugar se encuentra la tarea de **planificación**. En ella se realizó la estimación de los días de trabajo que iban a transcurrir con la realización de cada tarea hasta la fecha límite (Dicha fecha se ha considerado el 13 de junio). Cabe destacar que es en este primer bloque de tareas cuando se realizó el diagrama de Gantt mostrado posteriormente (ver figura 54).

Más tarde daría comienzo el conjunto de tareas relacionadas con el **análisis**. En esta parte se planificaron las tareas de análisis de aplicaciones similares a la desarrollada, que estaban encaminadas a proporcionar la información necesaria para conocer si el desarrollo de la aplicación supondría algo innovador y útil en cuanto a las aplicaciones ya existentes se refiere. De la misma forma, se planificaron las tareas relacionadas con el análisis de alternativas para realizar la implementación del juego y, por último, la obtención de requisitos.

Posteriormente, antes de que la fase de análisis llegue a su fin, da comienzo la fase de **diseño** de la aplicación. En dicha fase se planifica analizar la arquitectura del sistema, identificando el tipo de aplicación a desarrollar y los componentes que la forman.

Mientras el proceso de diseño es llevado a cabo, se planifica **implementar y codificar** la aplicación. Se planificó en primer lugar la instalación del entorno de programación (Software necesario y conexión con Google App-Engine) y más tarde la conexión con Facebook. Una vez hecho esto, se dio paso a la codificación del juego y las interfaces que lo componen, dejando para el final la parte relacionada con el estilo y apariencia. Finalmente, se realizó la estimación de la fase de pruebas y la comprobación de los últimos errores y retoques.

Todos los conjuntos de tareas descritos anteriormente serán realizados de forma paralela a la presente documentación, completando los diferentes apartados del documento desde el primer día de trabajo. Además, esta parte de **documentación** engloba las tareas relacionadas con la edición de las diapositivas necesarias para la defensa del proyecto.

El diagrama de Gantt elaborado que refleja la planificación de las tareas mencionadas es el siguiente:

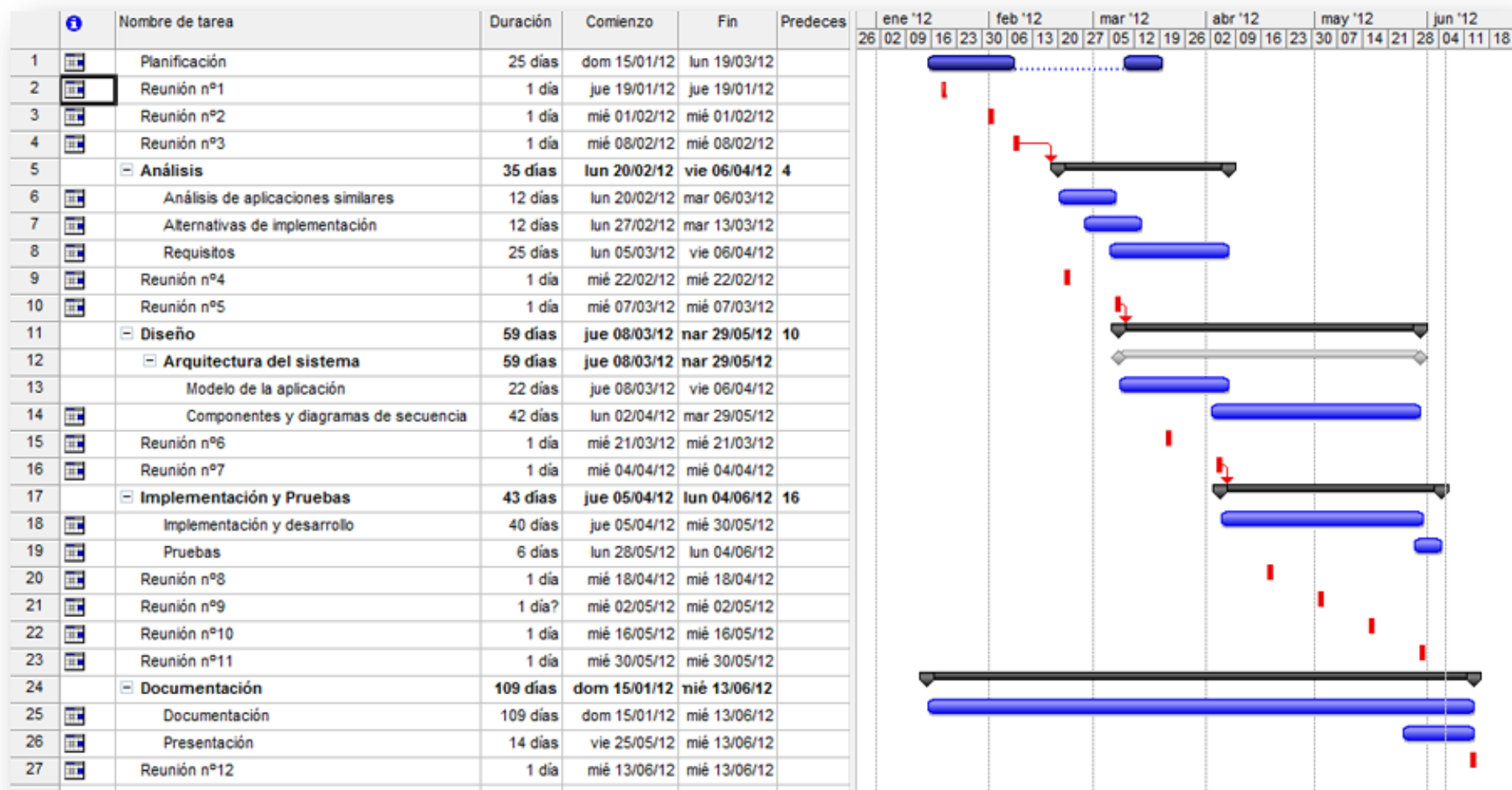


Figura 54: Planificación inicial, diagrama de Gantt

Cabe destacar que los puntos coloreados en rojo representan las mencionadas reuniones de seguimiento. Como se puede observar se realizan en algunas ocasiones de forma semanal y en otras de forma quincenal, dependiendo de la carga de trabajo planificada.

5.1.2. Planificación final

La planificación inicial ha sufrido varios cambios a lo largo del desarrollo del proyecto. No se ha tratado de un cambio muy significativo, pero sí ha supuesto la replanificación del tiempo de trabajo dedicado a varias de las tareas planificadas. Dichos cambios y los factores que han intervenido para variar la planificación inicial han sido los siguientes:

La fase de **planificación** se ha completado según previsto, realizando las tareas en el tiempo de trabajo especificado. Por otro lado, la fase de **análisis** ha sufrido algún que otro cambio. Esta ha tenido que ser ampliada una semana por el costoso análisis de Moodle como alternativa candidata a albergar la aplicación, ya que ha llevado más tiempo del previsto. Al igual que lo reflejado en la planificación inicial, la fase de **diseño** ha dado comienzo antes de dar por finalizada la de análisis. Esto ha sido realizado de esta manera para posibilitar el cambio de requisitos si era necesario para completar el diseño. Otra de las variaciones producidas ha sido el tiempo de trabajo dedicado a la fase de **implantación y codificación**. Para comenzar, esta fase se ha empezado a desarrollar de forma paralela a la de diseño, ya que, al contener tecnologías desconocidas previamente, se ha tenido que dedicar más tiempo del previsto para la implementación del entorno de programación (Google App-Engine) y las pruebas de conexión con Facebook. Cabe destacar que esta fase ha sido ampliada hasta el día 10 de junio, dando por finalizada la codificación de la aplicación el día 5 del mismo mes y la ejecución de las pruebas los 5 días restantes. Finalmente, la fase de **documentación** ha sido la que más cambios en cuanto a la planificación inicial ha sufrido. Esto ha supuesto que la redacción del presente documento diera comienzo el día 1 de mayo y no el primer día de trabajo, tal y cómo se refleja en la planificación inicial. Dicho cambio se ha producido por la carga de trabajo no planificado de las asignaturas propias del cuatrimestre, a las que se ha tenido que dedicar mucho más tiempo del previsto.

A continuación se muestra las tareas y el diagrama de Gantt que refleja la planificación final del proyecto:

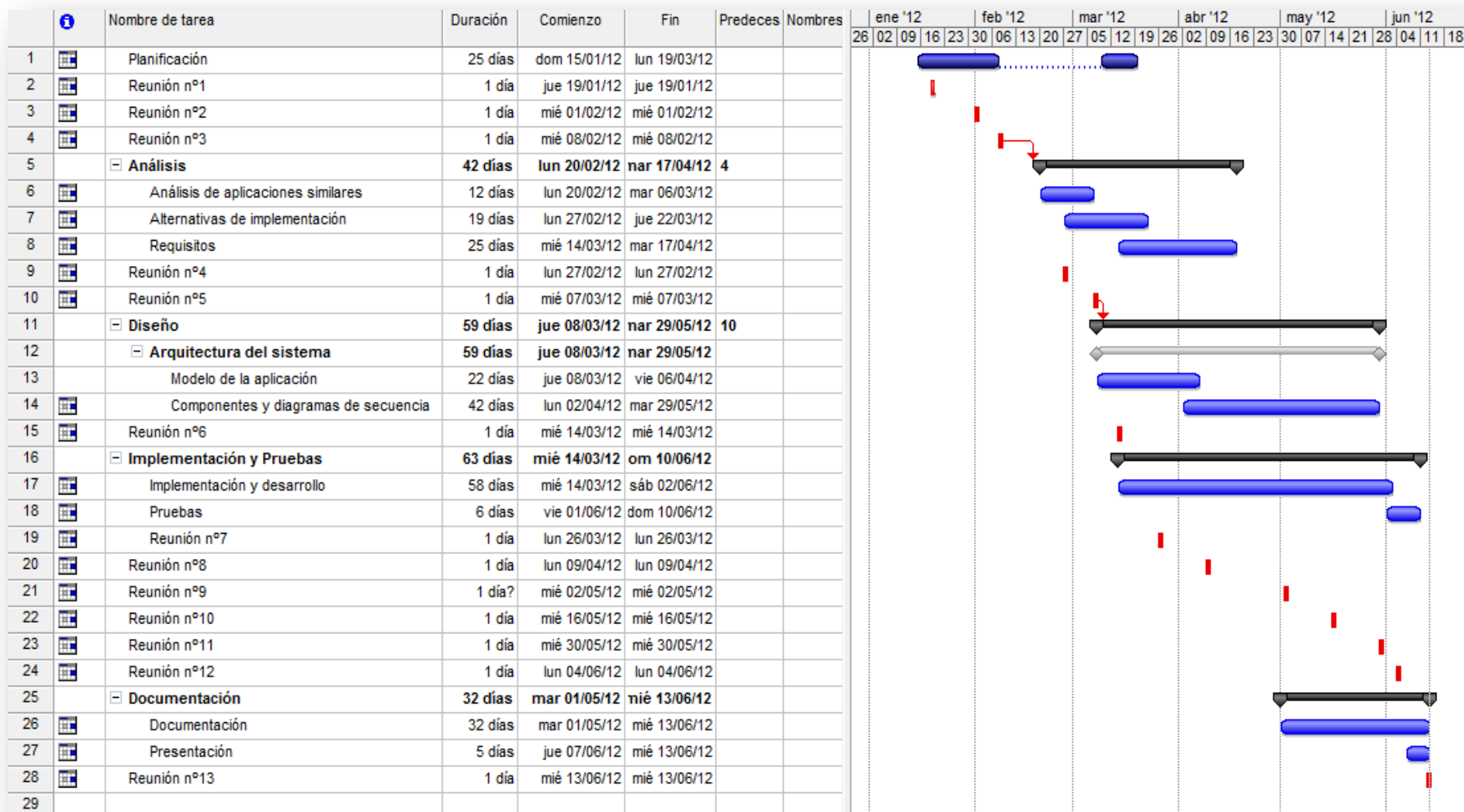


Figura 55: Planificación final, diagrama de Gantt

5.2. Medios técnicos empleados

A continuación se van a detallar todos los medios empleados para el desarrollo de la aplicación. El apartado se encontrará dividido en dos partes, la primera para especificar las herramientas *hardware* y la segunda para especificar las de tipo *software*. Cabe destacar que no se van a tener en cuenta los medios empleados por los profesores de Universidad que han dirigido el Trabajo de Fin de Grado.

5.2.1. Hardware

Como herramientas de tipo hardware se han empleado un ordenador de sobremesa para la implementación y codificación de la aplicación y un ordenador portátil para la elaboración de la documentación. Las características de dichos dispositivos se disponen en las siguientes tablas:

Dispositivo	PC de sobremesa
Procesador	Intel Core 2 Duo 2.13 GHz
Memoria Ram	4,00 GB
Disco Duro	500 GB
Tarjeta Gráfica	NVIDIA GeForce 8500 GT

Tabla 25: Herramienta hardware I

Dispositivo	Portátil HP Compaq 6720s
Procesador	Intel Celeron M 1.73 GHz
Memoria Ram	1Gb
Disco Duro	75Gb
Tarjeta Gráfica	Mobile Intel 965 Express Chipset Family

Tabla 26: Herramienta hardware II

5.2.2. Software

A continuación se va a mostrar una lista elaborada con los nombres de aquellas herramientas de tipo *software* empleadas para llevar a cabo el desarrollo de la aplicación. Dicha lista contendrá los siguientes campos: Tipo de software, nombre de la herramienta y página web donde se puede encontrar.

Tipo de software	Nombre de la herramienta	Sitio Web
Sistema Operativo	Windows 7Ultimate	http://emea.microsoftstore.com/es/es-ES/Microsoft/Windows/Windows-7
Procesador de textos	Microsoft Office Word 2010	http://emea.microsoftstore.com/es/es-ES/Microsoft/Office
Edición de diapositivas	Microsoft Office PowerPoint 2010	http://emea.microsoftstore.com/es/es-ES/Microsoft/Office
Entorno de programación	Eclipse IndigoServiceRelease 1	http://www.eclipse.org/downloads/
Control de versiones	Dropbox	https://www.dropbox.com/
Hoja de cálculo	Microsoft Office Excel 2010	http://emea.microsoftstore.com/es/es-ES/Microsoft/Office
Editor de imágenes	Adobe Photoshop Cs6	http://www.adobe.com/cfusion/tdrc/index.cfm?product=photoshop&loc=es_es
Conversor PDF	PDF Creator v1.2	http://www.pdfforge.org/download
Visor PDF	Adobe Reader X 10.1	http://get.adobe.com/es/reader/
Creación de diagramas de secuencia	Visual Paradigm for UML Enterprise edition	http://www.visual-paradigm.com/download/vpuml.jsp
Creación de diagramas de Gantt	Microsoft Office Project 2010	http://emea.microsoftstore.com/es/es-ES/Microsoft/Office
Creación de diagramas de clases	Star UML v5.0	http://staruml.sourceforge.net/en/

Tabla 27: Herramientas software

5.3. Análisis económico

A continuación se van a detallar todos los costes relacionados con la elaboración del proyecto. Primero se establecerán los costes estimados, especificando los de Recursos Humanos, los del hardware y software empleado y los costes indirectos. Posteriormente, se especificarán todos los costes reales, realizando una comparación con los estimados para obtener la diferencia existente entre ellos.

Finalmente, se va a realizar un análisis de las posibles formas de venta de la aplicación. Se mostrarán todos los resultados económicos resultantes indicando el grado de viabilidad que supone dicha venta.

5.3.1. Estimación de costes

Para estimar el coste total que supone el desarrollo de este proyecto se va a proceder a realizar el análisis de cuatro tipos de costes diferenciados. El primer tipo será el coste **personal**, donde se verá reflejado el precio que cada uno de los componentes de trabajo supone para el proyecto. Por otro lado, se indicarán los costes que suponen las herramientas de tipo **hardware y software** empleadas para el desarrollo de la aplicación, especificando precios de licencias, vida útil de la herramienta, número de unidades empleadas etc. Dentro de este tipo de coste no se incluirán los que suponen los medios empleados por los profesores de Universidad que han dirigido este proyecto en cuestión. Finalmente se especificarán los **costes indirectos**, que vendrán dados por el consumo de luz e internet producidos a lo largo del desarrollo del proyecto. Todos estos datos darán como resultado un precio final estimado del proyecto, que más tarde será sometido a comparación con los costes reales.

- **Costes de Personal**

Para realizar los cálculos de este tipo de coste se ha tenido en cuenta en primer lugar que el número de días laborables por mes es 20, que las horas de trabajo por día planificadas al principio del proyecto son 5 y que la duración del proyecto son 6 meses (Enero – Junio). Cabe destacar que el precio por hora de cada ingeniero informático ha sido obtenido de la hoja de cálculo obtenida como guía (26€/h).

Días/Mes	20
Meses	6
Horas/Día	5

Tabla 28: Coste de personal estimado I

Concepto	Horas	Coste ³⁰ (€/h)	Coste (€)
Ingeniero informático	600	26	15.600

Tabla 29: Coste de personal estimado II

³⁰ El coste mencionado ya incluye los valores del IRPF y Seguridad Social

- **Costes de Hardware**

Concepto	Unidades	Precio unitario (€)	Vida útil estimada (meses)	Tiempo de uso (meses)	Coste para el proyecto (€)
Portátil HP Compact	1	550	48	6	68,75
PC sobremesa	1	780	48	5	81,25
				Total	150

Tabla 30: Coste de hardware estimado

- **Costes de Software**

Concepto	Coste licencia (€)	Vida útil estimada (meses)	Tiempo de uso (meses)	Coste para el proyecto (€)
Windows 7 Ultimate	229	36	6	38,16
Microsoft Office 2010	139	12	6	69,5
Adobe Photoshop cs6	161	48	3	10,06
Visual Paradigm for UML Enterprise Edition	1130	48	1	23,54
Microsoft Office Project 2010	775	12	1	64,58
			Total	205,85

Tabla 31: Coste de software estimado

- **Costes indirectos**

Concepto	Precio mensual	Tiempo de uso	Coste para el proyecto
Luz	60	6	360
Conexión a internet	40	6	240
		Total	600

Tabla 32: Costes indirectos estimados

- **Costes totales estimados**

Estimación costes totales	
Costes de personal	15.600,00 €
Costes de Hardware	150,00 €
Costes de Software	205,85 €
Costes indirectos	600 €
Total	16.555,85 €

Tabla 33: Costes totales estimados³¹

5.3.2. Costes reales

A continuación se detalla el coste real del proyecto. Al igual que se realizó anteriormente con los costes estimados, se indicará el precio final resultante a partir de los costes personales, de *hardware*, *software* y costes indirectos. Cabe destacar que en todos los tipos de coste se añadirá el estimado y la diferencia entre ambos.

- **Costes de Personal**

Como aclaración a las horas diarias dedicadas hay que mencionar que se ha trabajado 5 horas a tiempo completo desde el comienzo del proyecto (10 – enero) hasta el 10 de abril. Por el contrario, debido a la intensidad y carga de trabajo de las asignaturas del cuatrimestre, se disminuyeron las horas de trabajo desde el mes de abril hasta el día 10 de mayo, pasando a dedicar una media de 2,5 horas por día de trabajo en el proyecto. Por el contrario, una vez finalizado el período de exámenes, se pasó a dedicar 8 horas de trabajo diarias el último mes antes de la fecha límite.

Dedicación	Horas diarias dedicadas	Meses	Horas totales
Tiempo completo	5	4	400
Tiempo completo	8	1	160
Tiempo parcial	2,5	1	50
Total			610

Tabla 34: Costes reales de personal I

³¹ Todos los costes indirectos, de *hardware* y *software* ya incluyen el valor del IVA

Answer2Pass – Juego de evaluación interactiva de alumnos

Concepto	Horas	Coste ³² (€)	Coste RRHH inicial (€)	Coste RRHH real (€)	Diferencia (€)
Ingeniero informático	610	26	15600	15860	-260

Tabla 35: Costes reales de personal II

- **Costes de Hardware**

Concepto	Unidades	Precio unitario (€)	Vida útil estimada (meses)	Tiempo de uso (meses)	Coste real para el proyecto (€)	Coste estimado para el proyecto (€)	Diferencia (€)
Portátil HP Compact	1	550	48	1	11,4583333	68,75	57,29
PC sobremesa	1	780	48	5	81,25	81,25	0
Total					92,7083333	150	57,29

Tabla 36: Costes reales de hardware

- **Costes de Software**

Concepto	Coste licencia (€)	Vida útil estimada (meses)	Tiempo de uso (meses)	Coste real para el proyecto (€)	Coste estimado para el proyecto (€)	Diferencia (€)
Windows 7 Ultimate	229	36	6	38,1666	38,1666	0
Microsoft Office 2010	139	12	1	11,5833	69,5	57,91
Adobe Photoshop Cs6	161	48	4	13,41666667	10,0625	-3,35
Visual Paradigm for UML Enterprise Edition	1130	48	1	23,54166667	23,5416667	0
Microsoft Office Project 2010	775	12	1	64,583333	64,58333333	0
Total				151,29166	205,8541667	54,56

Tabla 37: Costes reales de software

³² El coste mencionado ya incluye los valores del IRPF y la Seguridad Social

- **Costes indirectos**

Concepto	Precio mensual	Tiempo de uso	Coste real para el proyecto (€)	Coste estimado para el proyecto (€)	Diferencia (€)
Luz	60	6	360	360	0
Conexión a internet	40	6	240	240	0
Total			600	600	0

Tabla 38: Costes reales indirectos

- **Coste real total**

Concepto	Costes totales reales	Costes totales estimados	Diferencia
Costes de personal	15.860 €	15.600 €	-260 €
Costes de Hardware	92,71 €	150,00 €	57,29 €
Costes de Software	151,29 €	205,85 €	54,56 €
Costes indirectos	600 €	600 €	0 €
Total	16.704,00 €	16.555,85 €	-148,15 €

Tabla 39: Costes reales totales ³³

Como se observa en la tabla anterior, los costes estimados para la realización del proyecto ascienden a **16.555,85 €**, mientras que los costes que realmente ha supuesto han sido de **16.704,00 €**. Por lo tanto se observa una diferencia de **148,15 €**, que supone que los fondos reales empleados han superado a los estimados. Cabe destacar que no es una cantidad muy elevada en relación con el precio total, por lo que se puede decir que la estimación no ha resultado del todo satisfactoria pero tampoco se han sufrido pérdidas significativas.

³³ Todos los costes indirectos, de *hardware* y *software* ya incluyen el valor del IVA

5.4. Análisis de la forma de venta de la aplicación

Principalmente han sido dos las opciones candidatas como forma de venta de la aplicación. La primera se basa en la venta íntegra de la aplicación a un cliente, y la segunda en la explotación de la misma por medio de la integración de *banners* publicitarios.

5.4.1. Venta a un cliente

Las condiciones de venta suponen el traspaso total de la aplicación y sus derechos de autor, permitiendo al comprador realizar todo tipo de cambios en la misma para modificar su apariencia y funcionalidad. El porcentaje de beneficio que se quiere obtener es del 30%, por lo que el precio de venta debe ascender a **21.522,611€**, obteniendo así un beneficio de **4.966,76€**. Este precio ha sido calculado a partir del precio estimado del proyecto indicado en la tabla 39. Cabe destacar que el 30% de beneficio se ha estimado en base a la falta de competidores en el mercado, ya que el número de aplicaciones de este tipo implementadas en redes sociales es muy bajo.

Una vez sabido el precio real del proyecto (calculado en la figura XX), se obtiene que el beneficio obtenido al vender la aplicación asciende a **4.818,61€**, lo que supone una pérdida de **148,15€** en cuanto al beneficio estimado inicialmente. No obstante, se sigue consiguiendo un beneficio muy similar al estimado, ya que se trata de una pérdida poco significativa. Por ello se llega a la conclusión que esta primera opción de venta es viable.

Por otro lado, partiendo de que se trata de un nuevo cliente con el que nunca se ha llevado a cabo ningún tipo de negocio, se tiene en cuenta un 20% de riesgo de crédito. Dicho riesgo se tiene en cuenta por el posible incumplimiento de las obligaciones contractuales por parte del cliente. Por lo tanto, si ese porcentaje es aplicado al beneficio obtenido al vender la aplicación, resulta una cantidad de **3854,88€** de beneficio final. Cabe destacar que teniendo en cuenta el riesgo mencionado se produce una pérdida de **1111,88€**, lo que disminuye de forma significativa el beneficio estimado. No obstante esto sigue sin suponer la inviabilidad de esta primera opción.

5.4.2. Integración de banners publicitarios

Para estimar las ganancias producidas a través de la publicidad en forma de *banners* integrados en la aplicación, se va a tomar como referencia la cantidad monetaria indicada en campañas publicitarias llevadas a cabo por Impresiones Web³⁴, sitio web dedicado a la compra y venta de publicidad. Específicamente se va a tener en cuenta un precio CTM (*Cost Per Mille*) de 0,04€, que será cobrado una vez se alcancen las mil impresiones del *banner* en la aplicación.

Para comenzar el análisis se estima que 100 alumnos matriculados en una asignatura van a tener acceso a la aplicación, lo que significa que esta contará aproximadamente con una cifra de 100 usuarios registrados. También se estima que al ser una aplicación implementada en una red social dichos usuarios accederán a la aplicación una media de 2 veces al día. Otra condición inicial a especificar es la realización de los cálculos en torno al 30% de beneficio esperado y la posibilidad de integrar 2 o 4 *banners* de publicidad. Los cálculos realizados son los siguientes:

- Integración de dos *banners* publicitarios

Estimación beneficio/día	
Alumnos registrados	100
Visitas/día	10827
CTM (€)	0,4
Nº Banners	2
CTM total (€)	0,8
Precio/día (€)	0,16
Tiempo transcurrido hasta obtener el beneficio estimado	
Beneficio esperado (€)	21.522,611
Días	134.516,32
Meses	4.483,88
Años	373,66

Tabla 40: Integración de dos banners publicitarios

³⁴ Para más información acceder a www.impresionesweb.com

Answer2Pass – Juego de evaluación interactiva de alumnos

- Integración de cuatro *banners* publicitarios

Estimación beneficio/día	
Alumnos registrados	100
Visitas/día	10827
CTM (€)	0,4
Nº Banners	4
CTM total (€)	1,6
Precio/día (€)	12,9924
Tiempo transcurrido hasta obtener el beneficio estimado	
Beneficio esperado (€)	21.522,611
Días	67.258,16
Meses	2.241,94
Años	186,83

Tabla 41: Integración de cuatro banners publicitarios

Una vez realizado el análisis, se puede llegar a la conclusión de que debido a la baja cantidad de usuarios registrados en la aplicación esta forma de venta queda totalmente descartada por la cantidad de años necesarios para llegar a obtener el beneficio esperado. No obstante, en el apartado 6.3 dedicado a las líneas futuras, se proponen diversas mejoras que pueden incrementar de forma muy significativa el número de usuarios de la aplicación, lo que supondría una mayor viabilidad en cuanto a la integración de *banners* se refiere.

6. Conclusiones

A continuación se disponen las conclusiones obtenidas una vez finalizado el proyecto. En primer lugar se establecerán las conclusiones relacionadas con el proyecto en sí y posteriormente las conclusiones a nivel personal.

6.1. Conclusiones del proyecto

Se ha llevado a cabo un Proyecto de Fin de Grado encaminado al desarrollo de una aplicación web. El principal objetivo establecido fue implementar dicha aplicación en forma de juego de tipo ‘pregunta-respuesta’, que los profesores puedan ofrecer a sus alumnos para evaluarlos de una forma distinta a las convencionales. El profesor desempeñará la función de administrador de la aplicación, moderando y creando nuevas preguntas, gestionando peticiones de acceso y controlando los tableros de juego. El alumno se ocupará de jugar partidas, recorriendo un tablero respondiendo preguntas y creando otras, que serán recibidas por otros alumnos facilitando así la interacción entre los mismos. Todas las puntuaciones obtenidas quedarán dispuestas en varios rankings, que los profesores podrán utilizar para realizar la evaluación de los alumnos. Además de los objetivos iniciales mencionados se añade otro relevante obtenido tras el proceso de análisis, la implementación de la aplicación en una red social, en concreto Facebook. De esta forma, los alumnos podrán hacer uso de la aplicación mientras navegan de forma habitual por su red social, dando la posibilidad de compartir puntuaciones (aumentando en cierto grado la competitividad) y recibir notificaciones por parte de los profesores.

Todos los objetivos y funcionalidades explicados anteriormente han sido completados con éxito al finalizar el periodo de trabajo planificado al inicio del proyecto, obteniendo resultados muy satisfactorios y pasando todas las pruebas de aceptación propuestas. Esto se ha conseguido llevando a cabo las tareas de una forma muy parecida a la planificada, sufriendo variaciones poco significativas. De la misma forma, cabe destacar que el conjunto de requisitos se ha ido ampliando y modificando según el avance del diseño y la implementación, dando como resultado una aplicación completa, de fácil uso y con peculiares funcionalidades que permiten al usuario disfrutar de ella y aprovecharla para alcanzar los objetivos establecidos.

6.2. Conclusiones personales

Cuando leí la propuesta y la descripción del proyecto por primera vez, fue inmediata la acción del típico *'copy-paste'* y la promoción del mismo dentro de mi lista de proyectos más interesantes. Esta lista principalmente recogía proyectos que contaran con alguna parte de desarrollo, principalmente aplicaciones útiles que otros pudieran usar en un futuro. Mis dudas en cuanto a la búsqueda del trabajo de fin de grado se disiparon cuando acabé la reunión del mismo, había encontrado un proyecto basado en el desarrollo de una aplicación muy interesante cuyos objetivos llamaban enormemente mi atención.

La primera impresión que tuve sobre este proyecto no ha cambiado en ningún momento a lo largo del desarrollo del mismo, todo lo contrario, cada vez que aparecían nuevos requisitos e ideas se incrementaba mi interés por llevarlos a cabo, pensando en cada rato libre como poder mejorar lo pensado y qué cosas nuevas añadir. Con todas esas ideas se fue construyendo poco a poco una estructura de proyecto bastante más compleja e interesante de lo que yo había imaginado. Prueba de ello es el primer modelo de prueba implementado de la aplicación, donde a través de una sencilla interfaz los alumnos visualizaban y respondían todas las preguntas propuestas por el profesor y este moderaba las cuestiones enviadas por los alumnos; en comparación con la versión final, un auténtico juego implementado en una red social donde los alumnos juegan obteniendo y compartiendo diferentes tipos de puntuaciones mientras los profesores editan y administran el contenido del mismo. Cabe destacar que llegó un momento en el que eran tantos los cambios producidos, que pensé que iba a suponer una gran dificultad llevar a cabo su desarrollo. Este pensamiento cambió cuando, después de dedicar muchas horas de trabajo y esfuerzo, todo iba saliendo poco a poco según lo previsto.

La finalización de la aplicación y la memoria ha supuesto alcanzar un alto grado de satisfacción personal, ya que he conseguido sacar adelante un proyecto interesante y entretenido pasando por todas las fases de desarrollo software estudiadas en la carrera. Dicha satisfacción se debe de la misma forma al esfuerzo y horas empleadas para resolver problemas surgidos en la aplicación. Estos problemas y otros factores, como la carga de trabajo en las últimas asignaturas de la carrera, han ido transformando de alguna forma el proyecto en un reto personal, que finalmente ha finalizado de forma satisfactoria por el esfuerzo empleado y la ayuda recibida por mis tutores Jorge y José María.

Por otro lado, la realización de este proyecto me ha brindado la oportunidad de adquirir nuevos conocimientos no aprendidos a lo largo de la carrera. Por un lado, he aprendido las ventajas que supone el uso de Google-App Engine como herramienta de desarrollo web, que incluye el aprendizaje relacionado con JDO, una herramienta de almacenamiento de datos completamente nueva para mí que he aprendido a implementar y desarrollar de forma muy satisfactoria. Por otro lado, los conocimientos adquiridos que considero más relevantes han sido los relacionados con la

implementación y conexión de la aplicación con Facebook. A partir de ahora, no sólo sabré con más detalle cómo se comporta Facebook de forma interna en cuanto al intercambio de datos se refiere, si no que seré capaz de desarrollar aplicaciones en dicha red social obteniendo así los beneficios que esta brinda a sus usuarios. Cabe destacar que dichos conocimientos no sólo sirven para ampliar los ya aprendidos a lo largo de toda la carrera, si no que pueden ser útiles en un futuro para el desarrollo de propias aplicaciones o adecuados para diversos puestos de trabajo.

6.3. Líneas futuras

A continuación se enuncian diferentes cambios a los que se podría someter la aplicación desarrollada en este proyecto, ampliando así su funcionalidad, apariencia, portabilidad o accesibilidad.

Uno de los cambios que se proponen como mejora de la asignatura es la futura traducción a varios idiomas, ampliando así la posibilidad de acceso a usuarios de otras nacionalidades o a personas de nuestro propio país que carezcan de conocimientos de inglés.

Por otro lado, otro de los cambios y en mi parecer uno de los más importantes, sería la ampliación del juego para poder ser disfrutado en un primer paso por alumnos y profesores de toda la universidad. Esto sería posible cambiando la gestión de preguntas y tableros por parte de los administradores, permitiendo seleccionar el tipo de tablero según la asignatura cursada por un grupo determinado de alumnos. De la misma forma, se debería modificar el sistema de puntuaciones dispuestas en los rankings de tal forma que pudieran ser filtradas por asignatura o curso. Como un segundo paso, también se podría ampliar el rango de usuarios aceptados en la aplicación a todos los usuarios de Facebook, o lo que es lo mismo, que todos los profesores del mundo entero puedan crear y editar partidas destinadas a sus alumnos. Este gran cambio, aunque parezca requerir numerosas modificaciones en la aplicación, sólo supondría implantar un sistema organizado de administradores, donde cada uno controle su propio banco de preguntas y los tableros de juego. Cabe destacar que este cambio supondría un aumento en la viabilidad de la forma de venta explicada anteriormente en el apartado 5.4.2. Integración de banners publicitarios.

Otro cambio ya menos significativo, sería la implementación de un generador de tableros que no sólo se encargara de aleatorizar la disposición de temas y comodines (Ya implementado en Answer2Pass), si no que fuera capaz de cambiar la forma de estos. De esta forma se ofrecería al usuario un abanico más extenso de caminos a seguir para llegar a completar la partida.

De la misma forma, en cuanto al servidor empleado se podría optar por las mejoras de pago que Google ofrece, consiguiendo así una mayor velocidad de

Answer2Pass – Juego de evaluación interactiva de alumnos

transmisión de datos y una mayor capacidad de almacenamiento suficiente para albergar muchos más datos de jugadores registrados.

En cuanto a la gestión de la aplicación por parte de los profesores, se podría dividir dicha gestión en dos partes diferenciadas: Una formada por un solo administrador ocupado de desempeñar tareas como la gestión del dominio o el mantenimiento de la base de datos, y otra formada por todos los profesores dedicados a gestionar las preguntas, la configuración de la partida y las peticiones recibidas.

Finalmente, otro cambio propuesto para mejorar la funcionalidad de la aplicación sería la ampliación de temas, nuevos tipos de pregunta y nuevos modos de juego. De esta forma, el usuario podrá contar con una mayor diversidad de preguntas planteadas (Actualmente el máximo número de temas es 3), con preguntas de diferente modalidad como 'Rellena los huecos' o 'Une con flechas' y con nuevos modos de juego que proporcionen al usuario un mayor entretenimiento e incrementen las ganas de jugar.

ANEXO I - Manual de usuario

A continuación se desarrolla el manual que debe seguir el usuario para realizar un uso correcto de la aplicación implementada. En dicho documento se identifican, con ayuda de imágenes detalladas, todos los elementos que forman las diferentes partes del juego. Esto se lleva a cabo tanto para la parte dedicada a los usuarios que posean el rol de alumno, como para los profesores que desempeñen la función de administrador.

1. Interfaz de inicio

Una vez la aplicación es iniciada desde la red social Facebook, ya sea desde el buscador de aplicaciones, desde la sección de aplicaciones favoritas o desde alguna notificación recibida; se muestra una interfaz cuyo principal objetivo es introducir al usuario al contexto de la aplicación, especificar los pasos necesarios para hacer uso de ella y permitir o denegar el acceso a la misma.

Los elementos que componen dicha interfaz son los siguientes:



Figura 56: Manual de usuario, interfaz inicial

Como se puede observar, una de las secciones de la interfaz de inicio está reservada con el fin de explicar al usuario los pasos necesarios para acceder a la aplicación de forma correcta. Esto se lleva a cabo de la siguiente forma:

Lo primero que un usuario debe hacer para poder disfrutar de Answer2Pass es enviar una petición de acceso a los administradores de la aplicación (Profesores). Esto es necesario para poder permitir únicamente el acceso a alumnos que el profesor pueda verificar de forma previa. Para ello, únicamente es necesario especificar el nombre completo, el número de identificación del alumno (N.I.A) y el grupo de clase al que pertenece. Esta petición pasará a formar parte de una lista de peticiones pendientes hasta que el administrador la dé por válida o la rechace. Cabe destacar que el usuario sólo podrá enviar una petición mientras alguna otra se encuentre en estado de espera, y sólo podrá acceder al juego si su petición ha sido aprobada por el administrador. Finalmente, el usuario tendrá que permitir a la aplicación acceder a su información básica y la posibilidad de envío de notificaciones y publicación de mensajes en el muro. Esto lo puede realizar con el botón habilitado para dicha acción

especificado en la imagen anterior. Una vez completados estos dos pasos (Siendo sólo necesarios una vez), sólo queda pulsar el logo del juego para poder acceder al mismo.

2. Parte dedicada al usuario de tipo alumno

En este apartado se procede a identificar todos los elementos que forman la interfaz de juego, las reglas y desarrollo de cada una de las partidas y los diferentes tipos de rankings existentes.

2.1. Interfaz inicial de juego

Cuando un usuario accede al juego desempeñando el papel de ‘Alumno’, se le mostrará en primer lugar la pantalla de juego. Dicha pantalla estará formada por cuatro elementos claramente diferenciados: Un menú de opciones, el marcador de puntuaciones, la sección dedicada al almacenamiento de fichas bonus y el tablero de juego. El menú dará la posibilidad al usuario de situarse en el tablero de juego, acceder a los rankings de puntuaciones, comenzar una nueva partida y acceder a la sección dedicada a las preguntas sin responder recibidas por otros alumnos. El marcador de puntuaciones será el encargado de almacenar, en cada una de las partidas de juego, el número de tiradas acumulado que el usuario ha empleado hasta cierto punto del juego y el tiempo que ha sido empleado para responder las preguntas propuestas. Por otro lado, la interfaz de juego contiene una parte dedicada al almacenamiento de fichas bonus, cuya obtención y funcionalidad será explicada más adelante. Por último, y como elemento más relevante de esta pantalla, se encuentra el tablero de juego por el que el usuario irá avanzando gracias a la respuesta de preguntas planteadas por los profesores. Dichos elementos quedan identificados en la siguiente imagen:

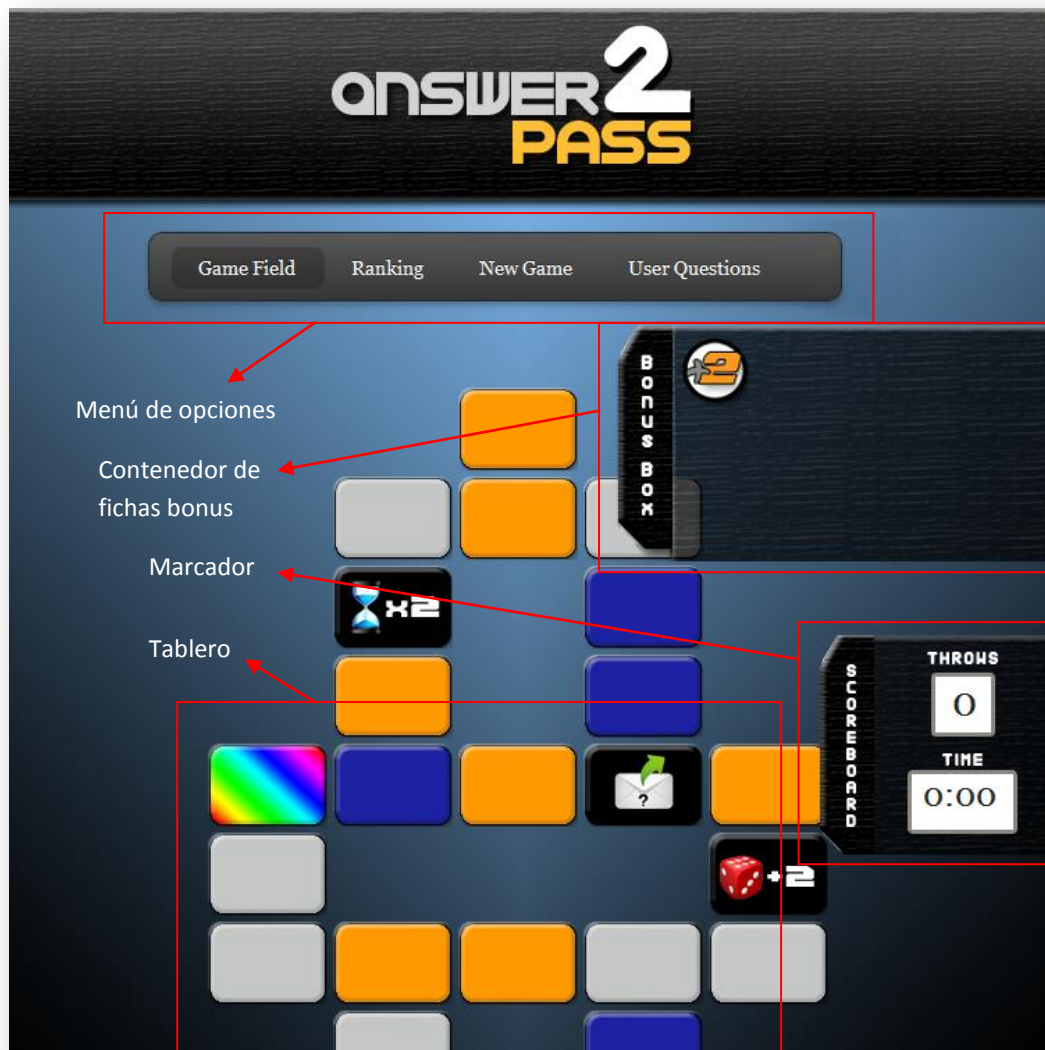


Figura 57: Manual de usuario, interfaz principal de juego

2.2. Jugar una nueva partida

Al dar comienzo la partida, la ficha de juego se sitúa en la primera casilla del tablero, y las puntuaciones del marcador y las fichas bonus son reiniciadas. El jugador debe hacer click en la casilla donde se sitúa la ficha de juego para que la pregunta sea mostrada. Las casillas se encuentran dispuestas de forma aleatoria por el tablero, diferenciándose por colores o comodines. Una vez se ha hecho click, la pantalla se oscurecerá mostrando la pregunta propuesta, cuyo tema vendrá definido por el color de la casilla. El tipo de la misma podrá ser 'Verdadero/Falso', 'Una sola respuesta' y 'Múltiples respuestas'. La pantalla de pregunta estará formada por dos contadores de tiempo: Una barra gráfica y un cronómetro. También se especificará la dificultad de la

pregunta por medio de estrellas (intervalo de dificultad [1-5]), el enunciado de la misma, y las respuestas posibles. Si el tiempo pasa y la pregunta no es respondida, el usuario se verá obligado a responder una nueva pregunta de ese tema sin posibilidad de avanzar por el tablero. Por el contrario, si el usuario responde a tiempo, el sistema le devolverá un mensaje indicando su acierto o fallo. Si ocurre lo primero, el usuario avanzará en el tablero un número proporcional de casillas al tiempo tardado en responder; si ocurre lo segundo, el usuario solo podrá avanzar una casilla. Cabe destacar que el avance por el tablero sólo está permitido hacia delante o de forma lateral, nunca hacia atrás. Una vez respondida la pregunta, el marcador se actualizará reflejando el acumulado de puntuaciones. Los elementos que forman la pantalla de preguntas se disponen en la siguiente imagen:



Figura 58: Manual de usuario, interfaz de respuesta a pregunta

Para avanzar más rápido por el tablero e interactuar con otros usuarios, el usuario puede hacer uso de varios comodines de juego. La funcionalidad de cada uno de los comodines viene descrita en las siguientes tablas:

Answer2Pass – Juego de evaluación interactiva de alumnos


Representación en el tablero	
Funcionalidad	Este tipo de comodín, llamado ‘Comodín sorpresa’, posee la funcionalidad de ofrecer al usuario un comodín aleatorio.

Tabla 41: Manual de usuario, tipo de comodín I


Representación en el tablero	
Funcionalidad	Este es uno de los comodines más relevantes del juego. Su funcionalidad reside en obligar al usuario a crear una nueva pregunta de juego para ser enviada a otro alumno aleatorio. Una vez hecho esto, el jugador avanzará 1 casilla en el tablero de juego.

Tabla 42: Manual de usuario, tipo de comodín II


Representación en el tablero	
Funcionalidad	Este es el comodín con la funcionalidad más simple de todos, permite al usuario responder una pregunta de tema aleatorio.

Tabla 43: Manual de usuario, tipo de comodín III


Representación en el tablero	
Funcionalidad	Además de aleatorizar el tema de pregunta, este comodín se ocupa de sumar 2 a la amplitud de tirada conseguida por el alumno al responder.

Tabla 44: Manual de usuario, tipo de comodín IV


Representación en el tablero	
Funcionalidad	Además de aleatorizar el tema de pregunta, este comodín se ocupa de multiplicar por 2 el tiempo límite de respuesta.

Tabla 45: Manual de usuario, tipo de comodín V

Como se ha mencionado en la tabla 42, el usuario se puede ver obligado en algún momento del juego a crear una nueva pregunta destinada a otro alumno. Esto será posible gracias a un editor de preguntas cuyos campos deben ser completados. Estos quedan especificados en la siguiente imagen:

The image shows a dark-themed user interface for creating a question. A white modal window is centered on the screen. Red arrows point from text labels to specific fields in the modal:

- Tipo de pregunta** points to the 'Type: Multi-Answer' dropdown menu.
- Wording:** points to the text input field for the question text.
- Respuestas posibles y correctas** points to the list of answer options, each with a checkbox.
- Dificultad** points to the 'Difficulty:' section, which includes a star rating.
- Tiempo** points to the 'Time (min):' input field, which currently shows the number '1'.

The modal also includes a 'Send' button at the bottom.

Figura 59: Manual de usuario, interfaz de creación de pregunta

Como se ha mencionado anteriormente, el usuario posee un contenedor de fichas bonus. Estas fichas pueden ser usadas para avanzar dos casillas por el tablero, y son conseguidas gracias a la respuesta satisfactoria a preguntas recibidas por otros usuarios. El usuario sabrá cuando tiene preguntas sin responder gracias a un icono con forma de exclamación mostrado junto a la opción del menú 'User questions'. De la misma forma, el alumno tendrá constancia de nuevas preguntas gracias a las

Answer2Pass – Juego de evaluación interactiva de alumnos

notificaciones enviadas a Facebook por parte de los administradores. La vista de preguntas sin contestar de otros alumnos es la siguiente:

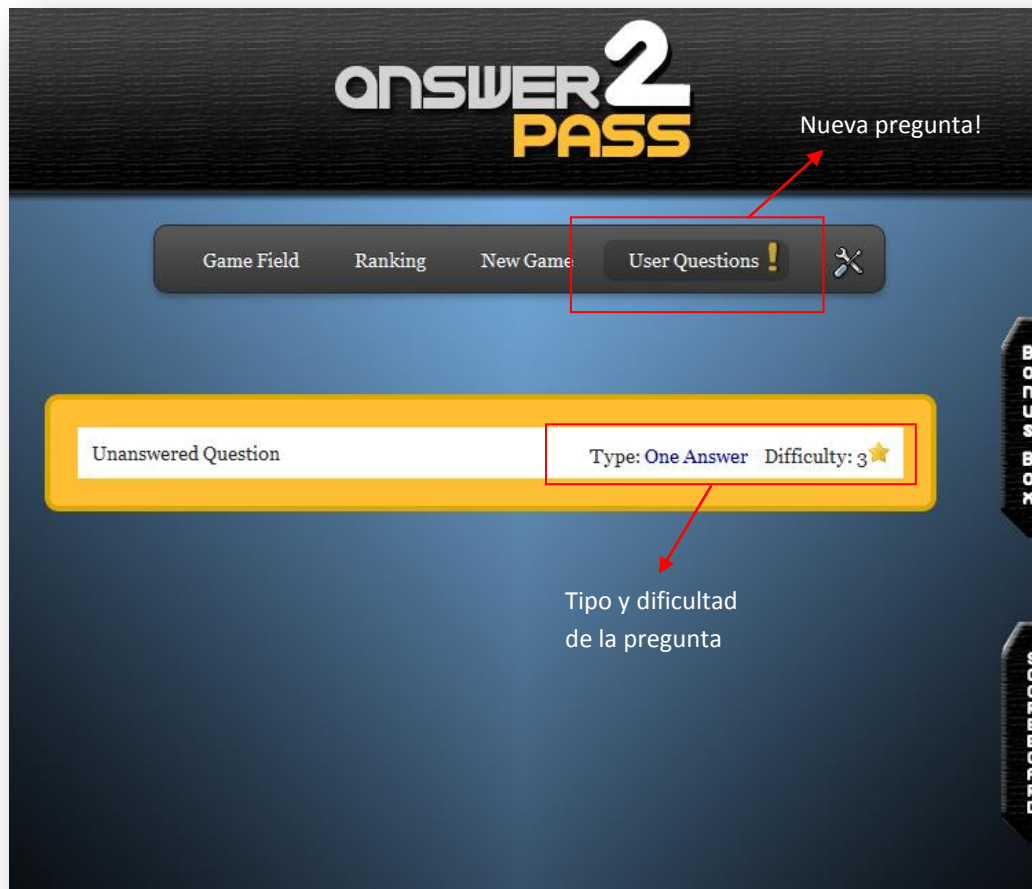


Figura 60: Manual de usuario, interfaz de preguntas recibidas por otros alumnos

Por último, una vez superado el tablero de juego, el alumno visualizará una pantalla que le mostrará la puntuación final obtenida. Desde aquí, el jugador podrá enviar dicha puntuación a su muro de Facebook, para compartirla así con todos sus contactos. Una vez se cierre esta ventana el sistema reiniciará de nuevo la partida.

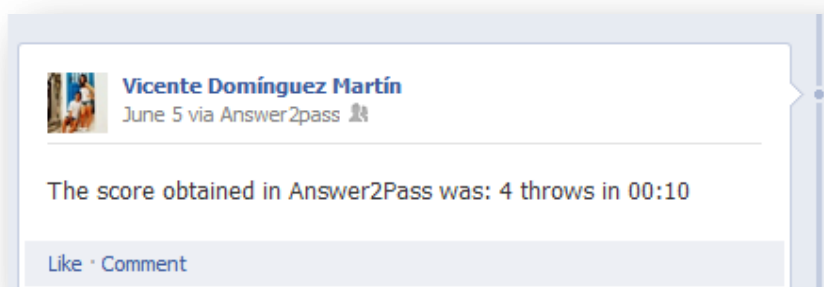


Figura 61: Manual de usuario, mensaje publicado en el muro de Facebook

2.3. Rankings

Los rankings de juego son las tablas donde quedan almacenadas las puntuaciones obtenidas por los usuarios. Estas serán visibles en todo momento por los profesores, que usarán para la correspondiente evaluación de los alumnos. Dichos rankings pueden ser de varios tipos:

Puntuación de la última partida: Este ranking denominado en el juego '*Score of the last game*', es el encargado de almacenar la puntuación obtenida en la última partida de juego. Los campos que muestra son: Nombre completo, cantidad de tiradas y tiempo empleado. Las puntuaciones estarán dispuestas en orden ascendente según el número de tiradas empleado. En caso de empate, promocionará el jugador que haya obtenido menor tiempo de respuesta.

Mejor puntuación: Este ranking denominado '*Best Score*', es el encargado de almacenar la mejor puntuación obtenida en las partidas de juego. Los campos mostrados y el orden de las puntuaciones son idénticos al ranking anterior.

Puntuación media: Este ranking denominado '*Average Score*', es el encargado de almacenar la puntuación media obtenida en las partidas de juego. Los campos que muestra son: Nombre completo, cantidad de tiradas, tiempo empleado y número de partidas. El orden de las puntuaciones es el mismo que los dos rankings anteriores.

Mejor preguntador: Este ranking denominado '*Best Questioner*', es el encargado de almacenar la puntuación de mejor preguntador obtenida por el usuario. Dicha puntuación se obtiene al enviar preguntas a otros alumnos. Si el profesor decide que la pregunta es adecuada y correcta, la puntuación aumentará según la dificultad de la pregunta planteada. Los campos que muestra este ranking son: Nombre completo y cantidad de puntos. Las puntuaciones estarán dispuestas en orden descendente según la cantidad de puntos acumulada.

3. Parte dedicada al usuario de tipo profesor

La parte de la aplicación visualizada por el profesor brinda la posibilidad de llevar a cabo tareas de control y moderación de la parte vista por los alumnos. De esta forma, aparte de poder acceder igualmente a los rankings y al tablero de juego, los profesores pueden gestionar las preguntas propuestas en el juego, moderar las enviadas por alumnos, gestionar el tablero de juego y aceptar o rechazar las peticiones de acceso recibidas. Todas estas opciones son explicadas con más detalle a continuación.

3.1. Gestión de preguntas de juego

La gestión de preguntas de juego es la opción que primero visualiza un administrador al entrar en la aplicación. Dentro de este apartado, el profesor puede crear nuevas preguntas de juego y eliminar las ya creadas. Para realizar lo primero, simplemente tiene que hacer click en el icono con forma de interrogación (Figura 64) y rellenar los campos dispuestos en un editor similar al mencionado en la parte del alumno (específicamente en el momento de crear nuevas preguntas para ser enviadas a otros alumnos), con la diferencia de los campos tema y vídeo (marcado como campo opcional). Cabe destacar que para rellenar de forma correcta este último campo, el profesor debe especificar como sigue en la siguiente imagen el identificador del vídeo obtenido de la página web 'Youtube' (nunca el enlace completo).

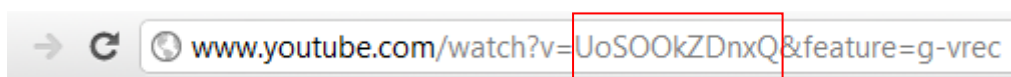


Figura 63: Manual de usuario, identificador de vídeo

Todas las preguntas creadas se dispondrán en una pila según el tipo de las mismas, la cual podrá ser filtrada según el tema, la dificultad y el tipo de la pregunta. Esta pila mostrará al administrador todos los campos propios de la cuestión. Dichas preguntas, pueden ser eliminadas haciendo uso del icono con forma de aspa roja situado en la esquina superior derecha de cada una de ellas. Dicho banco de preguntas se muestra en la siguiente imagen:

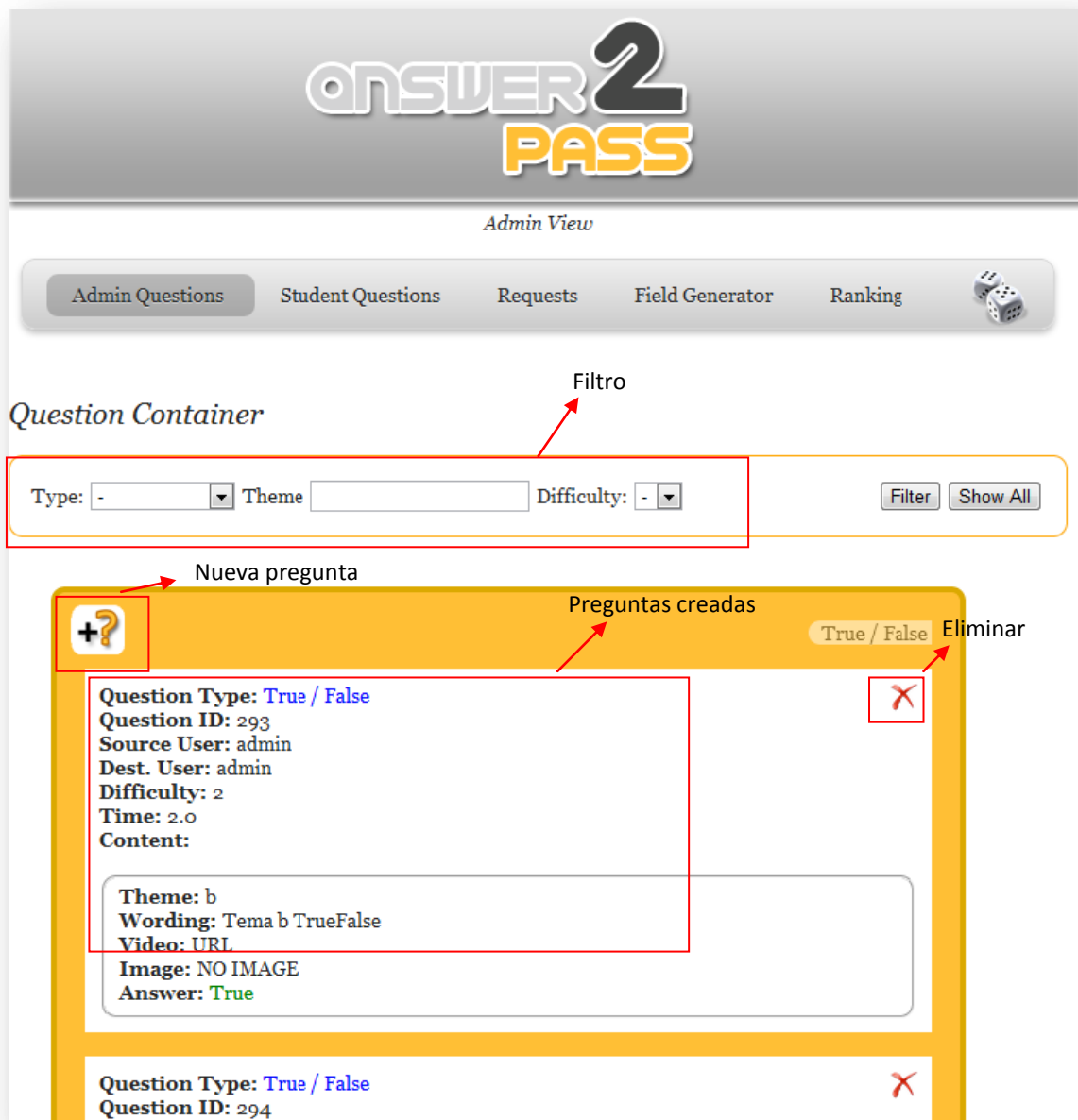


Figura 64: Manual de usuario, interfaz de gestión de preguntas de juego

3.2. Gestión de preguntas enviadas por los alumnos

Los profesores tendrán la obligación de corregir todas las preguntas enviadas por los alumnos antes de hacerlas llegar a sus correspondientes destinatarios. Dichas preguntas se dispondrán en dos contenedores casi idénticos al descrito en el apartado anterior, encontrados en la opción 'Student Questions' del menú de administrador. El

primero, almacenará las preguntas que se encuentran aún sin verificar, y el segundo, el conjunto de preguntas ya enviadas. Cabe destacar que el profesor tendrá la posibilidad de filtrar y eliminar las preguntas, además de poder notificar en el Facebook del alumno destinatario el envío una nueva. Para corregir las preguntas, el profesor únicamente debe pulsar el botón denominado 'Check And Send' habilitado para desempeñar dicha función. Automáticamente, las preguntas comprobadas y enviadas pasarán a formar parte del segundo contenedor y, a su vez, la puntuación de 'Mejor preguntador' del alumno se verá incrementada en función de la dificultad de la pregunta.

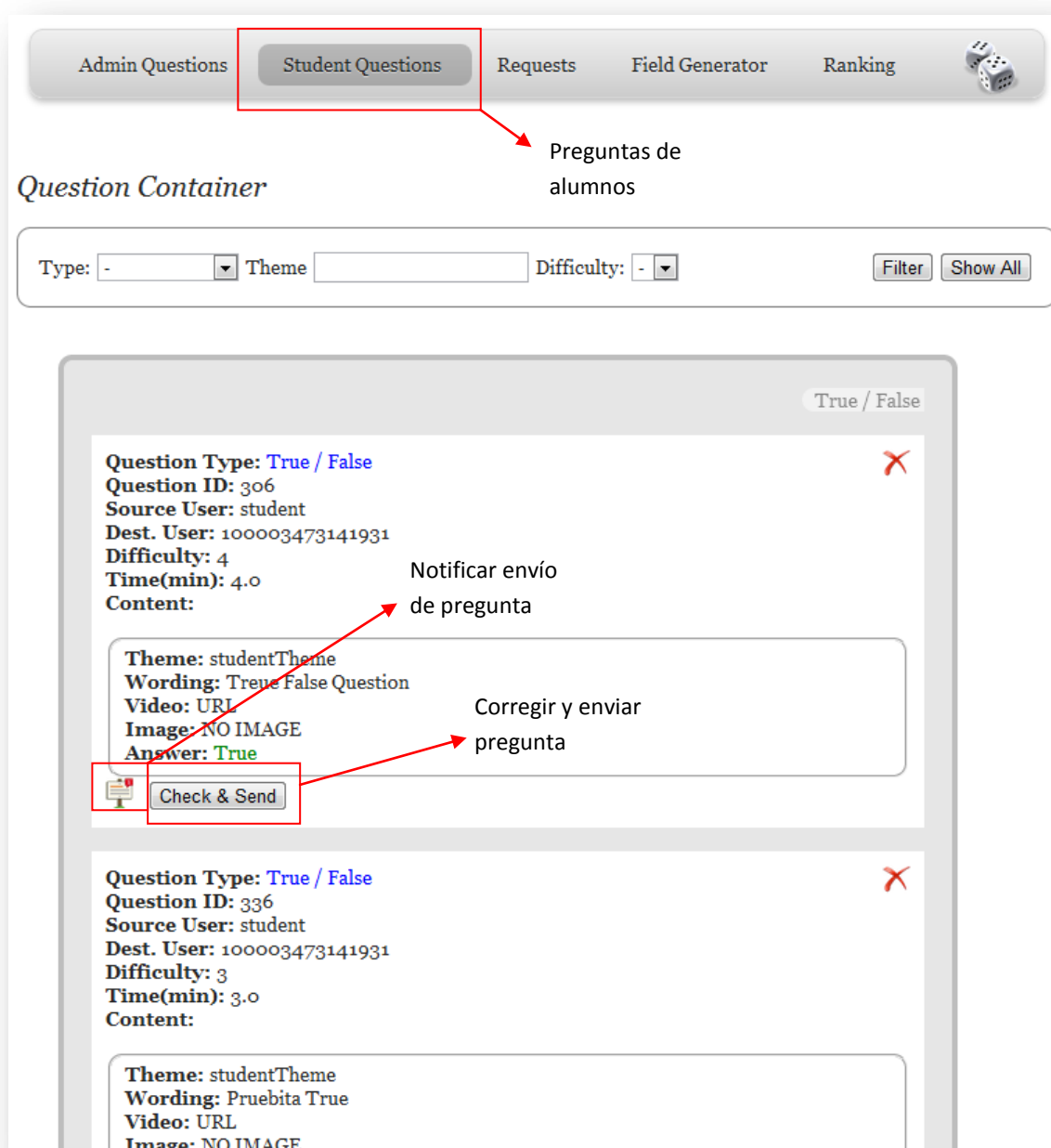


Figura 65: Manual de usuario, interfaz de gestión de preguntas de alumnos

3.3. Gestión de peticiones

Cada una de las peticiones de acceso enviadas por los usuarios que quieren hacer uso de la aplicación, deben ser moderadas por alguno de los administradores. Dichas peticiones serán dispuestas en una lista (Ubicada en la opción 'Requests' del menú de administrador) que mostrará el nombre completo, el grupo, el N.I.A y las opciones de aceptar, borrar o notificar. Las peticiones aceptadas pasarán a una nueva lista de usuarios registrados, permitiendo así el acceso a la aplicación. Las peticiones eliminadas se borrarán automáticamente de la lista, denegando así el acceso al juego. Cualquier acción anterior podrá ser remitida al usuario por medio de notificaciones en Facebook, que enviarán un mensaje el escrito por el profesor en el cuadro de texto habilitado para dicha acción. Los elementos descritos anteriormente quedan determinados en la siguiente imagen:

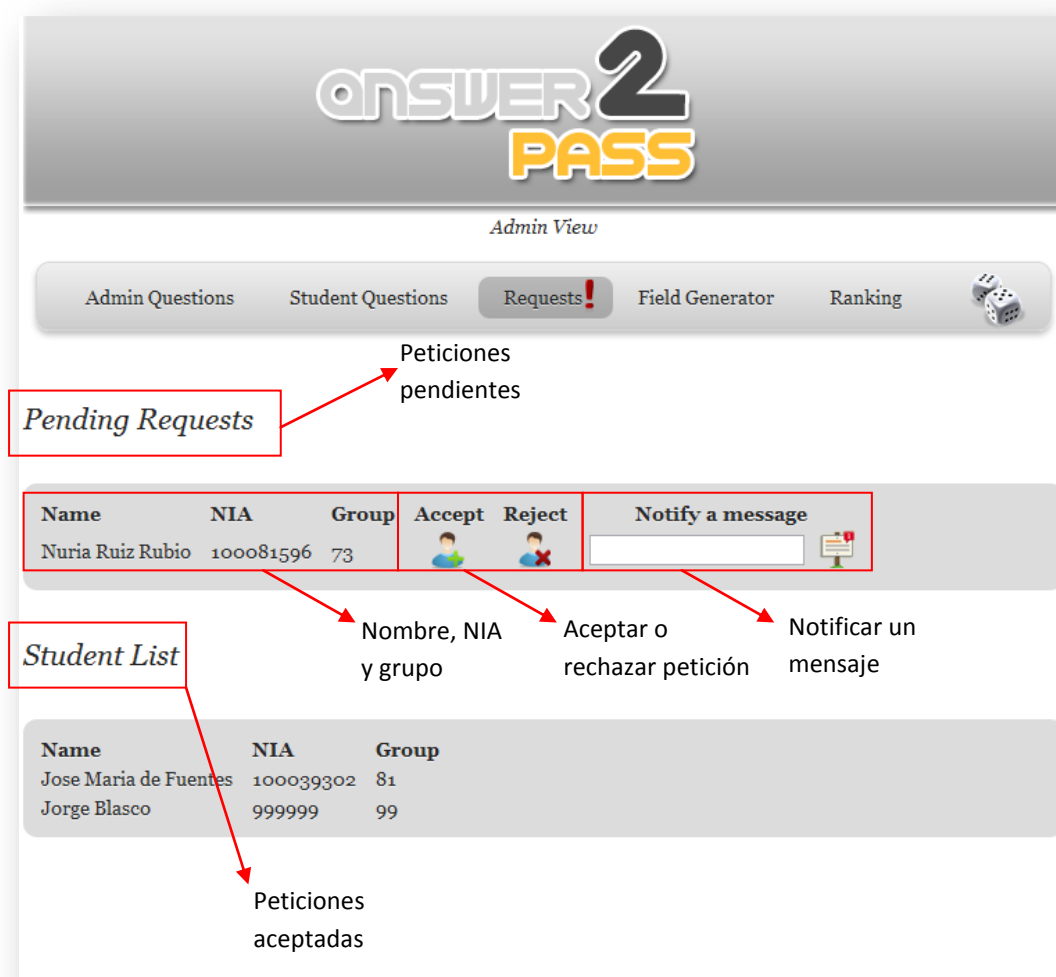


Figura 66: Manual de usuario, interfaz de gestión de peticiones

3.4. Gestión de Tableros

Los tableros de juego deberán haber sido generados previamente por los profesores. Esto es llevado a cabo en el apartado accesible mediante la opción '*Field Generator*' del menú de administrador. El profesor únicamente debe rellenar el nombre del tablero, especificar los temas de las preguntas y seleccionar el color correspondiente con el tema. Una vez hecho esto, debe pulsar el botón habilitado para comenzar la generación del tablero. Dicho generador se ocupará de disponer de forma aleatoria las casillas de tema y comodines a lo largo de todo el tablero, mostrando el resultado obtenido en una nueva pantalla. Los elementos necesarios para generar un nuevo tablero vienen especificados en la siguiente imagen:

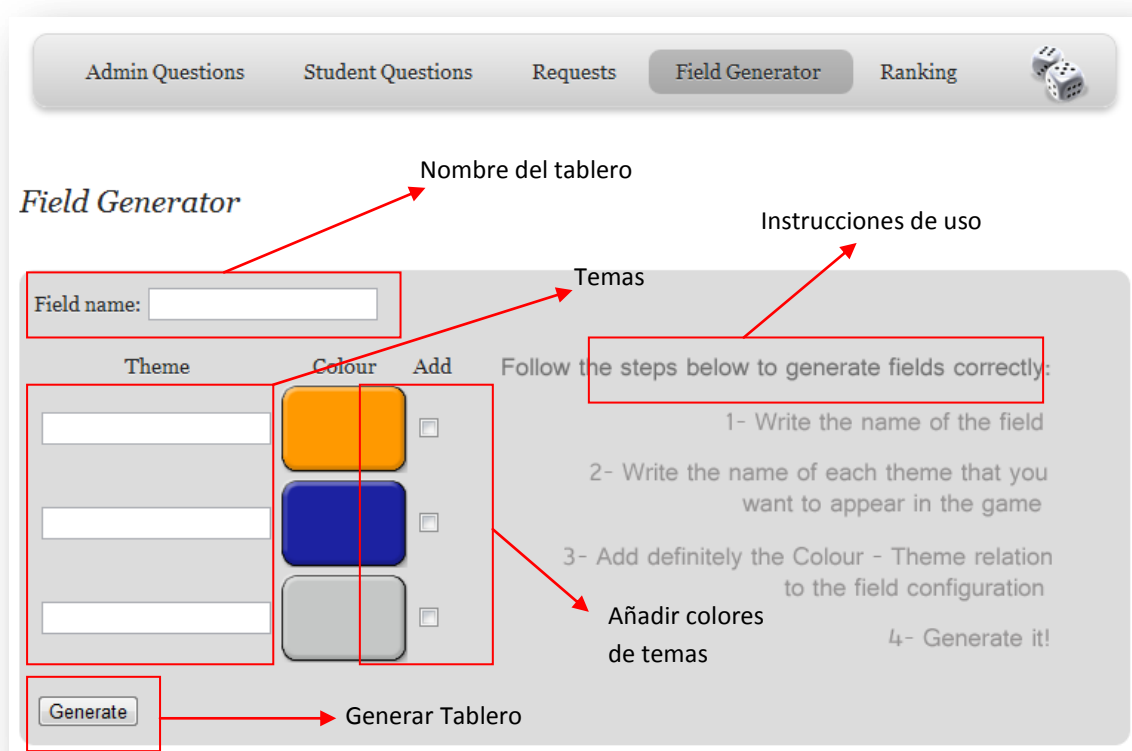


Figura 67: Manual de usuario, interfaz de gestión de tableros I

Todos los tableros generados pasarán a formar parte de una lista, la cual permitirá visualizar el nombre del tablero, los temas de los que trata y la eliminación y visualización del mismo. De la misma forma, la lista contendrá botones que permitirán seleccionar un tablero y establecerlo en el juego de todos los usuarios. Esto se puede llevar a cabo marcando el tablero correspondiente y pulsando el botón llamado '*Save*' para guardar el nuevo tablero de juego. Todos los elementos de dicha lista se muestran en la siguiente imagen:

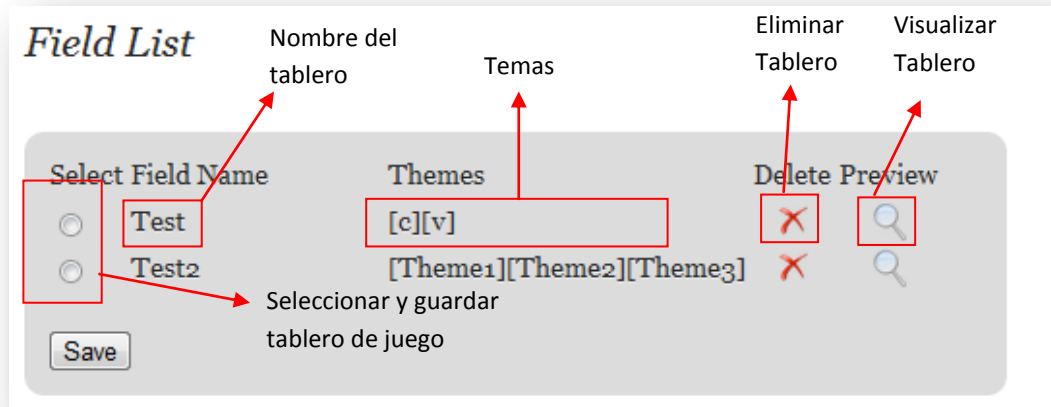


Figura 68: Manual de usuario, interfaz de gestión de tableros II

3.5. Visualización de Rankings y tablero de juego

Al igual que los alumnos, los profesores también pueden probar el tablero de juego y visualizar los rankings con las puntuaciones obtenidas por los alumnos. Estas acciones se pueden llevar a cabo haciendo uso del icono con forma de dados y la opción 'Rankings' dispuesta en el menú de administrador.

Fuentes de las imágenes utilizadas

- I. Apple. (n.d.). *Itunes Apple*. Retrieved Mayo 2, 2012, from <http://itunes.apple.com/es/app/trivial-pursuit-edicion-master/id456115098?mt=8>
- II. Computer History Museum. (n.d.). *Computer History Museum*. Retrieved Mayo 20, 2012, from <http://www.computerhistory.org/>
- III. Guttridge, N. (n.d.). *I, Science*. Retrieved Junio 2012, 10, from <http://www.isciencemag.co.uk/blog/are-you-a-social-network-junkie/>
- IV. Moodle. (n.d.). *Moodle*. Retrieved Abril 25, 2012, from http://docs.moodle.org/22/en/Quiz_module
- V. [V]Rishabh, T. (n.d.). *Wikipedia*. Retrieved Mayo 15, 2012, from http://commons.wikimedia.org/wiki/File:Facebook_popularity.PNG?uselang=es

Bibliografía

- 1- Amazon. (s.f.). *Amazon*. Recuperado el 5 de Marzo de 2012, de <http://aws.amazon.com/es/ec2/>
- 2- Chitnis, M., Tiwari, P., & Ananthamu, L. (s.f.). *Developer.Com*. Recuperado el 20 de Mayo de 2012, de <http://www.developer.com/design/article.php/2109801/Creating-Use-Case-Diagrams.htm>
- 3- Dentzel, Z. (s.f.). *Tuenti*. Recuperado el 25 de Febrero de 2012, de <http://blog.tuenti.com/ahora-tuenti-con-lengua-p/>
- 4- El Universal. (s.f.). *El Universal*. Recuperado el 25 de Febrero de 2012, de <http://www.eluniversal.com.mx/articulos/70485.html>
- 5- Facebook. (s.f.). *Facebook*. Recuperado el 5 de Marzo de 2012, de <https://developers.facebook.com/>
- 6- Garside, E. (s.f.). *HTML Drive*. Recuperado el 15 de Abril de 2012, de <http://www.htmldrive.net/items/show/818/clocks-timers-and-stopwatches-epiClock-jQuery-Plugin>
- 7- Gmarwaha. (s.f.). Recuperado el 5 de Mayo de 2012, de <http://www.gmarwaha.com/blog/2007/08/23/lavalamp-for-jquery-lovers/>

Answer2Pass – Juego de evaluación interactiva de alumnos

- 8- Google . (s.f.). *Google Code*. Recuperado el 15 de Marzo de 2012, de <http://code.google.com/p/facebook-java-api/>
- 9- Google. (s.f.). *Google*. Recuperado el 5 de Marzo de 2012, de <https://developers.google.com/appengine/docs/whatisgoogleappengine>
- 10- IABSpain. (s.f.). *IAB Spain*. Recuperado el 26 de Febrero de 2012, de <http://www.iabspain.net/ver.php?mod=noticias&identificador=123>
- 11- Impresiones Web. (s.f.). *Impresiones Web*. Recuperado el 25 de Mayo de 2012, de http://www.impresionesweb.com/precios_campanas_08.php
- 12- JQUERY. (s.f.). *JQUERY*. Recuperado el 30 de Marzo de 2012, de <http://jquery.com/>
- 13- Mestras, J. P. (s.f.). *Facultad de Informática. Universidad Complutense de Madrid*. Recuperado el 20 de Marzo de 2012, de <http://www.fdi.ucm.es/profesor/jpavon/poo/2.14.MVC.pdf>
- 14- Microsoft. (s.f.). *MSDN Microsoft*. Recuperado el 20 de Mayo de 2012, de <http://msdn.microsoft.com/es-es/library/dd409432.aspx>
- 15- Monge, S. (s.f.). *Taller 3D*. Recuperado el 2012 de Febrero de 25, de <http://www.tallerd3.com/archives/1506>
- 16- Moodle. (s.f.). *Moodle*. Recuperado el 27 de Febrero de 2012, de http://docs.moodle.org/22/en/Game_module
- 17- Moodle. (s.f.). *Moodle*. Recuperado el 27 de Febrero de 2012, de <http://moodle.org/about/>
- 18- Moodle. (s.f.). *Moodle*. Recuperado el 27 de Febrero de 2012, de <http://moodle.elearningbrothers.com/>
- 19- Ordenadores y portátiles. (s.f.). *Ordenadores y portátiles*. Recuperado el 1 de Junio de 2012, de <http://www.ordenadores-y-portatiles.com/que-es-arpanet.html>
- 20- Palanques, J. D. (s.f.). *DIXSO*. Recuperado el 6 de Abril de 2012, de <http://dixso.net/jquery/star-rating-con-jquery/>
- 21- Puro Marketing. (s.f.). *Puro Marketing*. Obtenido de <http://www.puromarketing.com/47/9901/millones-usuarios-todo-mundo-internet.html>
- 22- Pursuing Trivia. (s.f.). *Pursuing Trivia*. Recuperado el 1 de Marzo de 2012, de <http://www.pursuingtrivia.com/iphone/>
- 23- Rest FB. (s.f.). *Rest FB*. Recuperado el 15 de Marzo de 2012, de <http://restfb.com/>

Answer2Pass – Juego de evaluación interactiva de alumnos

- 24- Schiller, S. (s.f.). Recuperado el 15 de Mayo de 2012, de <http://www.schillmania.com/projects/fireworks/>
- 25- *Script Tutorials*. (s.f.). Recuperado el 20 de Abril de 2012, de <http://www.script-tutorials.com/animated-jquery-progressbar/>
- 26- Simón, A. (s.f.). *Cinco Días*. Recuperado el 25 de Febrero de 2012, de http://www.cincodias.com/articulo/empresas/tuenti-competencia-facebook-hecha-espana/20080411cdscdiemp_36/
- 27- The Real Time Report. (s.f.). *The Real Time Report*. Recuperado el 2012 de Febrero de 25, de <http://therealtimereport.com>
- 28- Universidad Carlos III de Madrid. (s.f.). *Universidad Carlos III de Madrid*. Recuperado el 25 de Mayo de 2012, de http://www.uc3m.es/portal/page/portal/organizacion/universidad_en_cifras/universidad_en_cifras_2011/estudiantes_2011
- 29- Wikipedia. (s.f.). *Wikipedia*. Recuperado el 25 de Febrero de 2012, de http://es.wikipedia.org/wiki/Red_social
- 30- Wikipedia. (s.f.). *Wikipedia*. Recuperado el 1 de Marzo de 2012, de http://en.wikipedia.org/wiki/Trivial_Pursuit
- 31- Wikipedia. (s.f.). *Wikipedia*. Recuperado el 20 de Marzo de 2012, de <http://en.wikipedia.org/wiki/Model%E2%80%93view%E2%80%93controller>
- 32- Wikipedia. (s.f.). *Wikipedia*. Recuperado el 1 de Junio de 2012, de <http://es.wikipedia.org/wiki/Internet>
- 33- Wikipedia. (s.f.). *Wikipedia*. Obtenido de <http://en.wikipedia.org/wiki/ARPANET>